Opinnäytetyö (AMK)

Tietojenkäsittelyn koulutusohjelma

Yrityksen tietoliikenne ja tietoturva

2014

Valtteri Holvitie

TUNKEUTUMISTESTAUS GEOLOKAATIOPALVELIMELLE

TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittely | Yrityksen tietoliikenne ja tietoturva

Toukokuu 2014 | 77 sivua

Ohjaaja: Jarkko Paavola

Valtteri Holvitie

TUNKEUTUMISTESTAUS GEOLOKAATIOPALVELIMELLE

Opinnäyte sisältyy osana Tekesin rahoittamaa White Space Test Environment for Broadcast Frequencies (WISE) -projektia, jossa on kehitetty kokeiluympäristö, joka tutkii kognitiivista radiota sekä verkkoa.

Opinnäytetyö alkaa hakkeroinnin määrittelyllä. Opinnäytetyö jatkuu tunkeutumistestauksen teoreettisella määrittelyllä. Tämän jälkeen tutkitaan sopivia vapaan tai avoimen lähdekoodin ohjelmistoja ja tekniikoita tunkeutumistestaamisen suorittamiseen.

Seuraavassa vaiheessa tarkoituksena on luoda WISE-projektissa käytettävää geolokaatiopalvelinta vastaava palvelin erilliselle testausjärjestelmälle ja löytää siitä mahdollisia tietoturva-aukkoja. Tunkeutumistestauksessa mukaillaan mahdollisen hyökkääjän käyttämiä tapoja, joita voidaan käyttää hyödyksi murtautuessa tietoverkkoihin ja -järjestelmiin. Opinnäytetyön tunkeutumistestaaja on niin sanottu White Hat -hacker, eli eettinen hakkeri. Eettinen hakkeri yrittää murtautua oman tai toimeksiantajansa tietoverkkoon ja -järjestelmiin haavoittuvuuksien testaamisen vuoksi ilman pahantahtoisia aikomuksia.

Kohdejärjestelmän tietoturvaa pystytään parantamaan tunkeutumistestauksen pohjalta tehtyjen havaintojen avulla. Tunkeutumistestauksen avulla pystytään lisäksi arvioimaan järjestelmän sen hetkisten suojausmekanismien toimivuus. Viimeisen vaiheen tavoitteena on tehdä onnistuneita hyökkäyksiä palvelimelle käyttämällä hyödyksi löydettyjä haavoittuvuuksia. Lopuksi on tarkoitus pohtia keinoja paikata haavoittuvuuksia ja kuinka niiltä voidaan suojautua.

ASIASANAT:

Tunkeutumistestaus, Haavoittuvuusskannaus, Palvelinympäristö, Geolokaatiopalvelin, Hyökkääminen, Paikkaus

BACHELOR´S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Business Information Technology | Business Data Communications and Information Security

May 2014 | 77 pages

Instructor: Jarkko Paavola

Valtteri Holvitie

PENETRATION TESTING FOR A GEOLOCATION SERVER

This thesis was carried out as a part of the Tekes-funded White Space Test Environment for Broadcast Frequencies (WISE) project, whoe objective is to examine the cognitive radio and network experiment environment.

The thesis begins with the definition of penetration testing, its phases and procedures. After this, it investigates suitable free or open source software and techniques to determine how far penetration testing can succeed.

The thesis next describes the process of creating a similar geolocation server that is used in the WISE project in a separate test environment and identifying potential security vulnerabilities. The penetration testing conforms to the real methods that a potential hacker could use to hack into the computer networks and systems. The penetration tester of the thesis is assumed to be a White Hat Hacker - also known as an ethical hacker. Ethical hackers try to break into their own or the client's information systems to perform testing without malicious intentions.

The target system security can be improved with the observations based on the penetration testing. Penetration testing allows for evaluating the system in addition to its current functionality of the security mechanisms. The objective of the last phase is to make successful attacks on the server by taking advantage of vulnerabilities found. The final phase of the thesis discusses ways to patch the vulnerabilities and to protect against them.

KEYWORDS:

Penetration Testing, Vulnerability Scanning, Server Environment, Geolocation Server, Patch

SISÄLTÖ

KUVAT

TAULUKOT

LYHENTEET JA SANASTO

Buffer Overflow Attack	Hyökkäys, jolla pyritään saamaan aikaan puskurin ylivuoto. Puskurin ylivuoto tapahtuu, kun ohjelma tai prosessi yrittää tallentaa enemmän tietoa kuin sen olisi tarkoitus. (Search Security 2014a.)
Cross-site scripting	Injektiohyökkäys, joka esiintyy usein WWW-sovelluksissa (Acunetix 2014a). Cross-site scripting tarkoittaa lyhyesti, että hyökkääjä voi jotakin kautta lisätä sivuille koodia, joka ei sinne kuulu. Se tunnetaan myös nimellä XSS-haavoittuvuus.
Demilitarisoitu alue	Aliverkko, joka on eristetty muista tietoverkoista (WiseGeek2014).
DNS	Domain Name System. Internetin nimipalvelujärjestelmä.
IDS	Intrusion Detection System. Tietoverkkoon asennettava järjestelmä, joka on ohjelmoitu tunnistamaan verkkoon suuntautuvat hyökkäysyritykset (Webopedia 2014a).
Käyttäjän manipulointi	Yritys saada yksittäinen käyttäjä paljastamaan tai antamaan pääsy turvattuihin tietoihin (Webopedia 2014b).
Madonreikähyökkäys	Hyökkäys, joka tunneloi tietoa toisesta verkosta toiseen (Answers 2014).
Palvelunestohyökkäys	Verkkohyökkäys, jossa estetään verkkosivuston tai -järjestelmän palvelujen saatavuus käyttäjille. Tunnetaan myös nimellä DDoS, eli Distributed Denial of Service. (Search Security 2014b.)
Rootkit	Ohjelmisto joka asentuu tietokoneelle, kun hyökkääjä on saanut sen hallintaansa (About 2014).
Spoofing	Spoofing-hyökkäys on yleinen termi hyökkäyksille, jossa hyökkääjä yrittää väärentää tietoverkossa kulkevaa tietoa. Esimerkiksi päästäkseen sisään johonkin järjestelmään tai lamauttaakseen sen. (ISS 2014.)
Tiedonvälittäjä	Käyttäjä, joka kerää tietoa ja tarjoaa tietoa muille organisaatioille (Social-Engineer 2014).
Traceroute	Työkalu, joka selvittää, mitä reittiä protokollan paketit siirtyvät määrättyyn koneeseen. (Webopedia 2014c)
Troijalainen	Haittaohjelma, joka tekeytyy viattomaksi sovellukseksi.
Vakoiluohjelmat	Ohjelmisto, joka kerää käyttäjän tietämättä häneltä tietoja. Tunnetaan myös nimellä ”spyware”.

Verkkourkinta	Huijausyritys, jolla käyttäjä yritetään saada antamaan itsestään yksityisiä tietoja. Tunnetaan myös nimellä ”phishing”. (Microsoft 2014.)

1 JOHDANTO

1.1 WISE-Projektin määritelmä ja tavoitteet

WISE (White Space Test Environment for Broadcast Frequencies) on projekti, joka tutkii radio- ja televisiotaajuuksien käyttöä kognitiiviradioiden avulla. Lisäksi WISE-projekti tutkii kognitiiviradioiden kaupallisen käyttöönoton mahdollisuuksia. (Wise 2014a.) WISE-projekti on osa Tekesin rahoittamaa Trial-ohjelmaa, joka tutkii kognitiivista radiota sekä verkkoa kokeiluympäristönä. Trial -ohjelma käynnistyi tammikuussa 2011 ja päättyy vuoden 2014 lopussa. Sen kokonaislaajuudeksi arvioidaan 30 miljoonaa euroa, josta Tekesin osuus on noin 14,5 miljoonaa euroa. Turun Ammattikorkeakoulun WISE-projektin yhteistyökumppaneita ovat muun muassa Turun yliopisto, Nokia, Digita, Fairspectrum ja viestintävirasto. (Tekes 2014.)

Kognitiivinen radio on älykäs radioteknologia, joka osaa käyttää kulloinkin sopivinta vapaata radiotaajuutta ja -verkkoa langattoman tietoliikenteen sujuvuuden varmistamiseksi. Jotta kognitiiviradioteknologian käyttö voitaisi kaupallistaa, täytyy varmistaa, etteivät TV- ja radiolähetykset häiriinny kognitiiviradioiden käytöstä. WISE-projekti etsii ratkaisua hyödyntämällä geolokaatiopalvelinta, josta kognitiiviradio voi tarkistaa, onko tietyssä kohdassa mahdollista toimia. (Wise 2014b.)

1.2 Tutkimusongelma ja tutkimussuunnitelma

Opinnäytetyön tavoitteena on löytää mahdollisia tietoturva-aukkoja WISE-projektissa käytettävästä geolokaatiopalvelimesta, joka sijaitsee internetissä. Tutkimusongelmana on geolokaatiopalvelimen turvallisuus, koska kognitiiviradiolaitteiden ja verkon toiminta ovat riippuvaisia siitä. Geolokaatiopalvelimen tietoturvaa ei ole perusteellisesti testattu, minkä seurauksena voi olla projektin kannalta ei-toivottuja seuraamuksia. Palvelimen

ylläpitäjien olisi tärkeä tietää, mitä keinoja ulkopuolisten taholta on käytetty hyökkäyksiin ja miten niiltä tulisi suojautua palvelimen turvallisuuden vuoksi.

Palvelimen sisällönhallintajärjestelmäksi on valittu Drupal. Drupal on niin sanottu sisällön hallintaan tarkoitettu runko, jonka avulla hallitaan muun muassa sivustojen ulkoasua, moduuleja ja käyttäjiä (Drupal 2014). Geolokaatiopalvelimen testaamisen käyttöjärjestelmäksi on valittu Debian. Jotta palvelin olisi mahdollisimman kevyt, Debian asennetaan ilman graafista ulkoasua. Debian on valittu sen ohjelmistojen vakauden, helpon asennettavuuden ja päivitettävyyden, tietoturvan, esiasetusten konfiguroinnin ja ilmaisen käyttämisen vuoksi.

Tunkeutumistestaamisessa käytetään muun muassa haavoittuvuusskannereita, jotka toimivat Windows 7-käyttöjärjestelmässä ja siihen luodulla virtuaalitietokoneella, johon on asennettu Kali Linux -käyttöjärjestelmä. Virtualisointiohjelmana käytetään VirtualBoxia. Lisätietoa tunkeutumistestauksesta löytyy esimerkiksi kirjoista The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy (Engebretson 2011) ja Advanced Penetration Testing for Highly-Secured Environments: The Ultimate Security Guide (Allen 2012).

1.3 Tavoite

Koska lupaa oikean geolokaatiopalvelimen levykuvan käyttämiseen ei saatu sen tietokannan sisältämän luottamuksellisen tiedon vuoksi, tarkoituksena on luoda siitä mahdollisimman tarkka kopio suojatulle ja erilliselle järjestelmälle. Tämän jälkeen tavoitteena olisi löytää siitä mahdollisia tietoturva-aukkoja. Tarkoituksena on mukailla ulkopuolisen hyökkääjän käyttämiä hyökkäysmenetelmiä ja murtautua onnistuneesti luodulle geolokaatiopalvelimelle. Hyökkäysten olisi tarkoitus jättää mahdollisimman vähän jälkiä muun muassa palvelimen lokijärjestelmään. Opinnäytetyön tutkimusote on konstruktiivinen tutkimus.

2 HAKKEROINNIN MÄÄRITTELY

2.1 Hakkeri

Hakkeri on henkilö, joka etsii ja käyttää hyväkseen tietokoneen tai tietokoneverkon heikkouksia. Hakkerille motivoivia tekijöitä voi olla useita, kuten esimerkiksi: hakkerointitaitojen parantaminen, protestointi tai haaste. Termi ”hakkerointi” yhdistetään usein kuvaamaan myös tietojärjestelmiin ja -verkkoihin ilkeämielisesti murtautuvaa henkilöä eli krakkeria. Yleisesti hakkeri ja krakkeri luokitellaan samaan luokkaan, mutta monien mielestä ne pitäisi erotella toisistaan (Catb 2014).

Hakkeri-termin voidaan määritellä tarkoittavan taitavaa tietokoneenkäyttäjää, joka tuntee tietokonejärjestelmien toiminnan syvällisesti ja joka hallitsee käyttämänsä työkalut ja laitteet erittäin hyvin. Hakkerille tietokone ei ole vain väline esimerkiksi taloudellisen hyödyn saavuttamiseen, vaan hakkeri on kiinnostunut hakkeroinnista myös oman mielenkiinnon vuoksi. Hakkereista on muodostunut hyvin negatiivinen kuva, sillä heidän ajatellaan yleisesti tekevän vain haittaa. Hakkeroinnilla kuitenkin tarkoitetaan myös aktiviteetteja, joiden tarkoitukset ovat hyviä (Iwriteiam 2014).

Krakkerilla tarkoitetaan henkilöä, joka yrittää murtautua tietojärjestelmään ilman järjestelmästä vastaavan osapuolen tai käyttäjien lupaa. Termiä käytetään myös henkilöstä, joka murtaa tietokoneohjelmien kopiosuojauksia ja salasanoja (PCTools 2014). Tietojärjestelmiin murtautuminen vaatii syvällistä työkalujen, ohjelmien tai ohjelmoinnin osaamista sekä kohdejärjestelmien käyttäjien varomattomuuden hyödyntämistä.

Hakkerin on mahdollista asentaa tietokoneeseen erilaisia haittaohjelmia, kun se on yhdistettynä verkkoon. Haittaohjelma voi muun muassa lähettää henkilökohtaista ja taloudellista tietoa ilman käyttäjän tiedostamista tai suostumusta. Hakkeri voi esimerkiksi

- kaapata käyttäjätunnuksia ja salasanoja
- varastaa rahaa tai avata luottokortteja tai pankkitilejä käyttäjän nimellä
- pyytää uusia PIN-koodeja tai luottokortteja
- tehdä ostoksia
- lisätä itselleen käyttäjätunnuksen luottotietoihin, millä on valtuudet käyttäjän kaikkiin toimiin
- väärinkäyttää sosiaaliturvatunnusta
- myydä tietoa muille osapuolille, jotka voivat käyttää niitä laittomiin tarkoituksiin (Webroot 2013).

2.2 Hakkerien luokittelu

White hat -hakkeri on henkilö, joka murtautuu tietojärjestelmiin hyvässä tarkoituksessa. Suuri osa white hat -hakkereista on tietoturvaeksperttejä, jotka tahtovat kokeilla oman tai toimeksiantajansa tietojärjestelmän ja verkon äärirajoja, kuten esimerkiksi salaus-algoritmeja ja suojauksia (Secpoint 2014). Joskus palkataan erikseen white hat -hakkereita testaamaan, miten haavoittuva sen hetkinen suojaus tietoverkoissa tai -järjestelmissä kokonaisuutena on. White hat -hakkeri tekee haavoittuvuusarviointeja sekä tunkeutumistestauksia. Heidät tunnetaan myös nimellä eettinen hakkeri.

Black hat -hakkerilla on usein ilkeämielisiä aikomuksia murtautuessaan johonkin tietojärjestelmään. Black hat -hakkeri tunnetaan myös nimellä krakkeri. Omaa hyötyään tavoitellen krakkeri voi käyttää hyödykseen eri teknologioita ilman oikeuksia, kuten esimerkiksi verkkoliikennettä, puhelinjärjestelmiä tai tietokoneita. Black hat -hakkerin pahansuovat tarkoitukset voivat vaihdella suuresti (Secpoint 2014). Esimerkkeinä voidaan pitää muun muassa tietoverkkorikoksia, identiteettivarkauksia, luottokorttipetoksia ja ilkivaltaa. Black hat -hakkeri voi käyttää hyödykseen kyseenalaisia taktiikoita, kuten esimerkiksi asentaa matoja kohdejärjestelmään tai käyttää hyödykseen haitallisia sivustoja saavuttaakseen tavoitteensa.

Grey hat -hakkeri on henkilö, joka omaa sekä white hat että black hat -hakkerin ominaisuuksia. Grey hat -hakkeri on hakkeri, joka ei ole tunkeutumistestaaja.

Hänen tavoitteenaan on etsiä internetistä esimerkiksi haavoittuvia järjestelmiä, joita voisi käyttää hyväksi. Onnistuttuaan murtautumaan esimerkiksi nettisivustolle grey hat -hakkeri ilmoittaa nettisivuston järjestelmän ylläpitäjälle sen haavoittuvuuksista (Secpoint 2013). Black hat -hakkereiden tavoin grey hat -hakkeri murtautuu mihin tahansa sivustolle ilman kehoituksia tai valtuuksia. Pientä maksua vastaan grey hat -hakkeri voi tarjoutua esimerkiksi korjaamaan haavoittuvat sivustot, joita hän paljasti.

Elite hacker on arvonimi, jolla tarkoitetaan keskimääräistä parempaa hakkeria. Arvonimi annetaan hakkerin erityisen hyvistä taidoista, ja muut hakkerit kohtelevat heitä usein kunnioittavammin. Elite hacker -nimityksellä tarkoitetaan hakkeria, jolla on vankka maine hakkerien keskuudessa, ja jolla on erityisen hyvä kyky harhauttaa hyökkäyksien jäljittäjiä. (Secpoint 2013)

Script kiddie on halventava nimitys hakkereista, joilla ei ole vielä kunnollista ammattitaitoa. He murtautuvat tietokonejärjestelmiin ilman IT-turvallisuuden tietämystä käyttäen muiden hakkerien tekemiä automaattisia työkaluja sekä ohjelmia (Secpoint 2013). Yleensä niin kutsuttu script kiddie ei juurikaan omaa tietoa käyttämiensä ohjelmien ja työkalujen konseptista tai niiden oikeista toimintatavoista.

Hacktivist on hakkeri, joka käyttää teknologiaa hyödykseen ilmoittaakseen esimerkiksi sosiaalisia, ideologisia, uskonnollisia tai poliittisia viestejä (Secpoint 2013). Yleensä niin kutsutut haktivistit muuttavan jonkun tietyn sivuston tietoja tai käyttävät esimerkiksi Denial-of-Service -hyökkäyksiä estämään jonkin palvelun toiminnan.

2.3 Hakkerointimenetelmät

Hakkerointimenetelmät jaetaan usein kahteen eri luokkaan: **syntaktiseen** ja **semanttiseen**. Syntaktiset hyökkäykset ovat yksinkertaisia ja suoraviivaisia. Niiden ajatellaan yleisesti olevan haittaohjelmia, jotka sisältävät esimerkiksi viruksia, matoja tai troijalaisia. (Crootof ym. 2011, 7.)

Semanttisella hyökkäyksellä tarkoitetaan hienovaraisempaa lähestymistapaa. Semanttinen hyökkäys on muunnelma oikeaa ja väärää tietoa, jota levitetään tarkoituksellisesti. Sen tavoitteena on käyttää käyttäjien luottamusta hyväksi, mikä täten voisi saada esimerkiksi tietokonejärjestelmän tuottamaan virheitä tai arvaamattomia tuloksia (Bhardwaj & Singh 2014). Väärän tiedon levittäminen suurelle määrälle ihmisiä onnistuu nopeasti eri mekanismien, kuten esimerkiksi sähköpostin ja sosiaalisen median kautta.

Hakkerointiyritykset voidaan jakaa tarkemmin eri ryhmiin:

1. verkkourkinta
2. väärennetyt sivustot
3. spoofing, eli hyökkäys, jossa hyökkääjä yrittää väärentää tietoverkossa kulkevaa tietoa
4. vakoiluohjelmat
5. sähköiset keskustelufoorumit
6. tiedonvälittäjät
7. troijalaiset
8. madonreikähyökkäykset (Bhardwaj & Singh 2014).

Kaappaus ja matkiminen

On olemassa monia eri tapoja, joita hakkeri voi käyttää matkiakseen muita käyttäjiä. Yleisin käytetty menetelmä on pahaa-aavistamattomien käyttäjien tietojen salakuuntelu eli eavesdropping. Tiedoilla yritetään saada haltuun käyttäjien tunnuksia, salasanoja ja muita käyttäjäkohtaisia tietoja. Hakkerit, jotka haluavat saada käyttäjätilejä haltuunsa, voivat käyttää troijalaisia apunaan. Troijalaiset on muun muassa suunniteltu varastamaan käyttäjien salasanoja (Search Security 2007). Hakkeri voi onnistua asentamaan troijalais-ohjelman käyttäjän koneelle esimerkiksi pikaviestiohjelman kautta. Pikaviestiohjelma voi kaiken lisäksi säilyttää käyttäjän tunnuksia erillisinä tiedostoina tietokoneella. Kun käyttäjä joko tahattomasti tai tarkoituksellisesti suorittaa ohjelman, troijalaisohjelma etsii käyttäjän salasanan koneelta ja lähettää löydetyt tiedot hakkerille.

Kaappaus on tietoturvahyökkäys, jossa hyökkääjä ottaa haltuunsa viestinnässä jonkin osapuolen, ja tekeytyy yhdeksi niistä. Eräässä kaappaustyylissä, joka tunnetaan myös nimellä ”man-in-the-middle attack”, hyökkääjä ottaa hallintaansa muodostetun yhteyden sen jo ollessa käynnissä. Tällöin hyökkääjä voi esimerkiksi kaapata viestejä vaihtamalla julkisen avaimen omaansa. (Search Security 2007). Hyökkääjä voi esimerkiksi käyttää ohjelmaa, jonka viestitys vaikuttaa olevan palvelimelta asiakkaalle ja toisinpäin. Tätä hyökkäystä voidaan käyttää viestien lukemiseen ja viestien muokkaamiseen, ennen kuin ne lähetetään uudelleen.

Toinen muoto kaappauksesta on selainkaappausohjelmat. Siinä käyttäjä ohjataan jollekin toiselle sivustolle, kuin johon hän on itse pyytänyt tai ollut siirtymässä. On olemassa kaksi erilaista tapaa ohjata selain eri kohteeseen käyttämällä DNS:ää (Domain Name System) eli internetin nimipalvelujärjestelmää hyödyksi. (Search Security 2007.)

Ensimmäinen tyyli on, että hyökkääjä saa käyttöönsä DNS-tiedot palvelimelta, ja onnistuu muokkaamaan niitä niin, että pyynnöt tietylle verkkosivulle ohjataan jonnekin toiselle sivustolle. Usein kohteena on väärennetty sivusto, jonka hyökkääjä on voinut itse luoda. Tämä antaa vaikutelman käyttäjälle, että sivusto on turvallinen (Search Security 2007). Edellä mainittuja hyökkäyksiä on vaikea estää, sillä ylläpitäjät hallitsevat vain omia DNS-tietojaan, eikä heillä usein ole hallintaoikeuksia muihin DNS-palvelimiin.

Toisentyyppisessä DNS-kaappauksessa hyökkääjä huijaa voimassaolevia sähköpostitilejä lähettämällä suuria määriä sähköposteja kohdetietokoneelle mukaillen oikeita teknisiä ja hallinnollisia kontakteja (Search Security 2007). Käyttäjä voi luovuttaa arkaluonteista tietoa vastatessaan kyseisiin sähköpostiviesteihin.

2.4 Suojautuminen

Suojautumista varten on selvitettävä, missä järjestelmän tai verkon mahdolliset tietoturva-aukot sijaitsevat. Löydetyt tietoturva-aukot olisi hyvä tunnistaa

useasta syystä. Ensinnäkin yleiset tietoturva-aukot ovat alueita, joihin organisaation tulisi puuttua nopeasti. Yleisten tietoturva-aukkojen ongelmana on se, että automatisoitujen työkalujen avulla kuka tahansa käyttäjä voi yrittää murtautua kohdeverkkoon tai -järjestelmään (Klevinsky ym. 2002, 29). Tarkoituksena on poistaa haavoittuvuudet ja oppia niistä lisää, jotta organisaatio kykenisi lieventämään altistumisen riskiä tulevaisuudessa.

Toiseksi, yleiset tietoturva-aukot ovat alueita, joihin organisaation tulisi kiinnittää huomiota tunkeutumistestaamisen aikana. Nämä haavoittuvuudet ovat hakkereille yleensä melko helppo tunnistaa ja hyödyntää (Klevinsky ym. 2002, 29). Järjestelmiin tunkeutuminen voi olla suhteellisen yksinkertaista ja helppoa, jos niitä ei ole päivitetty tai suojattu viimeisimpiä haavoittuvuuksia vastaan.

Järjestelmien pitäminen ajan tasalla on yhä vaikeampaa suuremmilla verkkokokonaisuuksilla, joilla on käytössä useampi eri käyttöjärjestelmä tai rajoitettu henkilöstön budjetti (Klevinsky ym. 2002, 29). Yksi hyvä keino pitää verkko turvallisena on seurata jatkuvasti löydettyjä haavoittuvuuksia ja päivittää järjestelmät niitä vastaan. Mitä enemmän reagoivia ylläpitäjiä on haavoittuvuuksien korjaamiseen, sitä turvallisempia järjestelmät ovat.

Konfiguraatiovirheet aiheuttavat riskejä, jotka mahdollistavat hyökkääjien tunkeutumisen järjestelmiin (Klevinsky ym. 2002, 30). Esimerkkejä konfiguraatiovirheistä ovat esimerkiksi tarpeettomien palvelujen pitäminen käynnissä, virheellisesti osoitetut tiedostojen oikeudet ja heikkolaatuiset salasanat.

Organisaatiot voivat vähentää konfiguraatiovirheitä luomalla standardeja oletusasetuksiin sekä asetusten hallitsemiselle (Klevinsky ym. 2002, 29-30). Lisäksi kunnollinen tunkeutumistestaus auttaa tunnistamaan monia reikiä asetuksista, jotka saattaisivat antaa hyökkääjälle pääsyn järjestelmiin.

Yleisesti ottaen ei ole olemassa keinoja sulkea kaikkia mahdollisia tietoturva-aukkoja verkossa. Nykyaikana mikä tahansa verkko ja järjestelmä on altis hyökkäyksille. Käyttämällä tarpeeksi aikaa ja rahaa pystytään kuitenkin suojautumaan tehokkaasti hyökkäyksiä vastaan. Tunkeutumistestaamisella ja

järjestelmien päivittämisellä pystytään kuitenkin sulkemaan noin 80-90 prosenttia kaikista tietoturva-aukoista. (Klevinsky ym. 2002, 29-30.)

Yksi järjestelmien ja verkkojen suurimmista haavoittuvuuksista ovat heikot salasanat. Tämä ongelma saadaan poistettua vain vahvemmilla autentikointijärjestelmillä, kuten esimerkiksi kaksivaiheisilla tunnistautumisella tai kertakäyttöisillä salasanoilla. Vaikka on olemassa tekniikoita turvallisten salasanojen muistamiseen, käyttäjät valitsevat usein helposti muistettavan salasanan. Tämä johtuu usein tietämättömyydestä tietoturvasta. (Klevinsky ym. 2002, 41.)

Käyttäjät usein helpottavat hakkereiden murtautumista entisestään valitsemalla yksinkertaisia salasanoja, kuten esimerkiksi nimiä, päivämääriä, urheilujoukkueita, tai muita merkittäviä seikkoja, jotka voivat olla helposti arvattavissa. Sama pätee käyttäjätunnuksiin. Järjestelmien ylläpitäjien on huomattu olevan aivan yhtä syyllisiä heikkoihin käyttäjätunnuksiin ja salasanoihin, kuin niin sanotut normaalitason käyttäjät. (Klevinsky ym. 2002, 42.)

Uudet salasanojen murtamiseen tarkoitetut ohjelmat ovat nykyään niin tehokkaita, että mikä tahansa sanakirjan sana pystytään murtamaan minuuteissa (Klevinsky ym. 2002, 41). Yksinkertaiset muunnelmat, kuten esimerkiksi takaperin käännetyt sanat, numeron lisääminen sanan alkuun tai loppuun ja muut vastaavanlaiset yksinkertaiset muunnelmat ovat lähes yhtä alttiita kuin sanat ilman muunnelmia.

Windows-käyttöjärjestelmissä tulisi aina olla päivitykset ajan tasalla. Windows on maailman myydyin käyttöjärjestelmä ja samalla myös hyökätyin kohde. Päivitykset minimoivat matojen, viruksien ja muiden haittaohjelmien mahdollisuutta murtautua järjestelmään (Western Carolina University 2014). Sama pätee myös ohjelmistopäivityksiin. Varsinkin selaimen ja muiden verkkopohjaisten sovellusten tulisi olla jatkuvasti päivitettyinä. Seuraavassa listassa on lueteltuna muutamia keinoja pitää järjestelmä suojattuna:

- Käytä virustorjuntaohjelmia ja pidä ne ajan tasalla.

- Älä avaa sähköpostia joka on lähetetty tuntemattomasta lähteestä.
- Käytä vaikeasti arvattavia salasanoja.
- Käytä palomuuria.
- Älä päästä muita käyttäjiä tietokoneellesi. Opettele tiedostojen jakamisen riskeistä.
- Katkaise internetyhteys, jos et käytä sitä.
- Varmuuskopioi tärkeät tiedostot.
- Lataa säännöllisesti tietoturvapäivityksiä.
- Tarkasta tietokoneesi tietoturva säännöllisesti.

3 TUNKEUTUMISTESTAUKSEN MÄÄRITTELY

Tunkeutumistestauksella tarkoitetaan tietokonejärjestelmän, verkon tai web-sovelluksen testaamista käytännössä. Tarkoituksena on löytää tietoturva-aukkoja, joita hyökkääjä voisi hyödyntää. Tunkeutumistestauksen ja hakkeroinnin karkea ero on siinä, mitä tiedolla tehdään ja onko se luvan varaista (Pen-tests 2014). Tunkeutumistestaaminen voidaan tehdä joko manuaalisesti tai automatisoimalla niitä sovellusten avulla. Prosessiin kuuluu muun muassa tietojen kerääminen kohteesta ennen testaamista tai tiedustelua, tunnistaa eri reittejä kohdeverkkoon tai -järjestelmään, yritystä murtautua sisään sekä havaintojen raportointia (Search Software Quality 2011). Suomessa tunkeutumistestaamista tekeviä yrityksiä ovat esimerkiksi Nixu, Silverskin ja Opsec.

On useita syitä, miksi tunkeutumistestaus suoritetaan. Yksi tärkeimmistä syistä on löytää ja korjata haavoittuvuuksia ennen kuin hyökkääjä löytää ne. Joskus organisaation IT-osasto on tietoinen raportoiduista haavoittuvuuksista, mutta se tarvitsee ulkopuolisen asiantuntijan virallisesti ilmoittamaan ylemmälle johtoportaalle niistä. Tämä tehdään usein siitä syystä, että johtoporras hyväksyisi tarvittavat resurssit kyseisten haavoittuvuuksien korjaamiseen (SANS 2006). Uuden järjestelmän testaaminen ennen käyttöönottoa on myös tärkeää tietoturvan vuoksi. Toinen syy tunkeutumistestaamiselle on antaa IT-osastolle mahdollisuus torjua mahdolliset hyökkäykset.

Tunkeutumistestaus on arvokasta monestakin syystä, kuten

1. määrittämään, mille hyökkäyksille tietoverkko tai -järjestelmä on altis
2. tunnistamaan korkeampien riskien haavoittuvuuksia, jotka johtuvat matalampien riskien haavoittuvuuksien yhdistelmistä
3. tunnistamaan haavoittuvuuksia, joita on vaikeaa tai mahdotonta havaita automaattisten verkko- tai sovellusskannausten avulla
4. arvioimaan onnistuneiden hyökkäyksien vaikutusta

5. testaamaan verkkoa puolustavien henkilöiden kykyä onnistuneesti havaita ja vastata hyökkäyksiin
6. tarjoamaan näyttöä siitä, että organisaation olisi lisättävä panostustaan turvahenkilöstöön ja teknologiaan (Penetration test 2014).

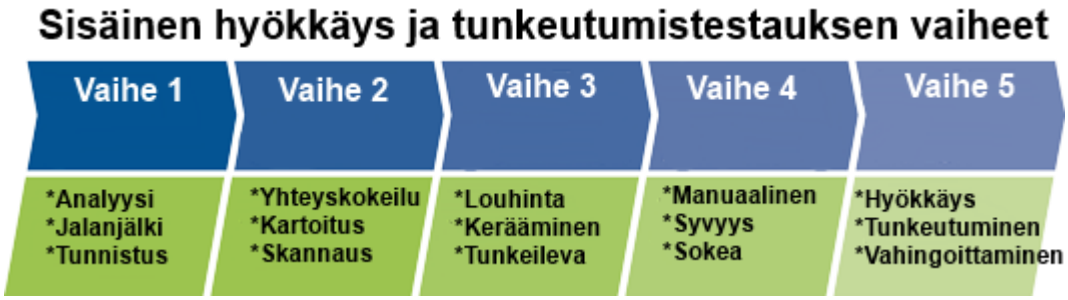
3.1 Tunkeutumistestaustyypit

Tunkeutumistestaus jaetaan kahteen eri tyyppiin: **ilmoitettuun** ja **ilmoittamattomaan**. Näiden kahden erottaminen toisistaan riippuu siitä, mitä halutaan testata (Klevinsky ym. 2002, 25 — 27). Testausta voidaan tehdä esimerkiksi verkon turvalaitteita tai verkon turvallisuuden henkilöstön reagointia varten.

Ilmoitettu testaaminen tarkoittaa yritystä päästä käsiksi ja saada haltuun etukäteen ilmoitettuja tiedostoja, tai murtautumista verkon järjestelmiin IT-henkilöstön avustuksella ja tuntemuksella. Tämän kaltaisella testauksella tarkastellaan sen hetkistä tietoturvainfrastruktuuria ja yksittäisiä järjestelmiä mahdollisten haavoittuvuuksien varalta (Klevinsky ym. 2002, 25 — 27). Ympäristön luominen, jossa organisaation tietoturvahenkilöstöt ovat osana tunkeutumistestausryhmää, mahdollistaa kohdennetut hyökkäykset kaikkein alttiimpiin kohdekoneisiin.

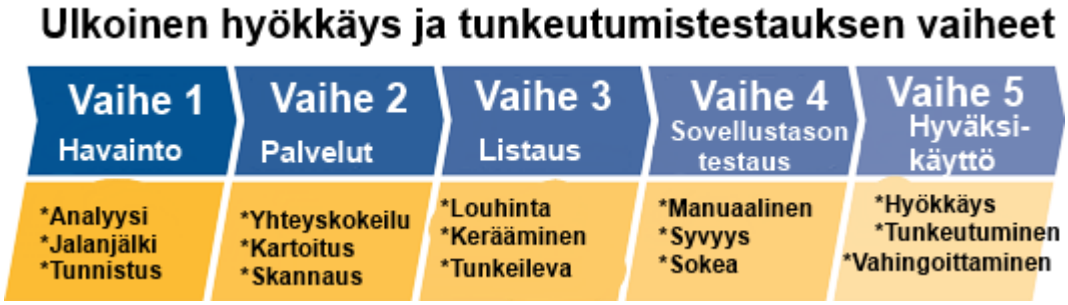
Ilmoittamaton testaaminen tarkoittaa, että vain yrityksen johto on tietoinen murtautumisyrityksistä. Murtautumisyritykset voivat olla esimerkiksi ennalta sovittujen tiedostojen haltuun ottaminen tai murtautuminen verkon järjestelmiin. Vastaavanlainen testaaminen tarkastelee sekä sen hetkistä tietoturvainfrastruktuuria että tietoturvahenkilöstön reagointikykyä (Klevinsky ym. 2002, 25 — 27). Jos tunkeutumisen havainnointi- ja hätätilannesuunnitelmat on luotu, testaukset voivat paljastaa mitkä tahansa heikkoudet niiden toteuttamisessa. Ilmoittamaton testaaminen tarjoaa testin koko organisaation turvallisuusmenetelmätapoihin, samalla kun turvallisuusinfrastruktuurillekin.

Aiemmin mainittujen lisäksi tunkeutumistestaus voidaan jakaa kahteen eri luokkaan: **sisäinen tunkeutumistestaus** ja **ulkoinen tunkeutumistestaus**. Sisäiset tunkeutumistestaukset yrittävät määritellä, mitä haavoittuvuuksia on järjestelmillä, jotka ovat saavutettavissa valtuutettujen verkkoyhteyksien tai kirjautumistunnusten kautta, ja jotka sijoittuvat organisaation toimialueelle. Sisäinen testi kannattaa tehdä esimerkiksi silloin, jos epäillään, että äskettäin irtisanottu työntekijä saattaa päästä käsiksi arvokkaisiin tietoihin. (Security Assessment 2014a.) Kuvassa 1 havainnollistetaan sisäisen hyökkäyksen ja tunkeutumistestauksen eri vaiheita.



Kuva 1. Sisäisen hyökkäyksen ja tunkeutumistestauksen vaiheet.

Ulkoinen tunkeutumistestaus on tarkoitettu tunnistamaan haavoittuvuuksia sen hetkisistä yhteyksistä, jotka on luotu organisaation kautta, kuten esimerkiksi palomuuri tai yhdyskäytävä. Ulkoinen tunkeutumistestaus on parempi vaihtoehto, jos ensisijaisena tavoitteena on esimerkiksi tunkeutumistestata palkkatietokannan tietoturvaa, johon on erillinen pääsy yrityksen internet-sivujen kautta (Security Assessment 2014b). Kuvassa 2 havainnollistetaan ulkoisen hyökkäyksen ja tunkeutumistestauksen eri vaiheita.



Kuva 2. Ulkoisen hyökkäyksen ja tunkeutumistestauksen vaiheet.

3.2 Tunkeutumistestauksen vaiheet

Esihyökkäysvaihe

Esihyökkäysvaihe voidaan jakaa kahteen eri vaiheeseen: **passiivinen ja aktiivinen tiedustelu.** Tunkeutumistestaaja tai hyökkääjä käyttää usein enemmän aikaa esihyökkäysvaiheeseen kuin itse hyökkäysvaiheen toteuttamiseen.

Passiivinen tiedustelu: Tässä vaiheessa tunkeutumistestaaja yrittää kerätä niin paljon tietoa yrityksestä kuin on mahdollista. Tarkoituksena ei ole luoda yhteyksiä kohdeverkkoon millään tavalla, vaan pysyä kohteelta täysin huomaamattomana (Vacca 2013, 531).

Passiivinen tiedustelu koostuu muun muassa seuraavista toiminnoista: web- ja ftp:n hakemistojen kartoitus, oikeuksien luokittelu, asiakirjojen tutkiminen (ainoastaan julkaistujen materiaalien tietojen kerääminen) sekä käyttäjän manipulointi eli social engineering (Melmeg 2014, 18).

Aktiivinen tiedustelu: Tässä vaiheessa pyritään profiloimaan ja kartoittamaan organisaation verkkoprofiili esimerkiksi myöhempää hyökkäystä varten. Tunkeutumistestaaja voi luoda yhteyksiä kohdeverkkoon, mikä kuitenkin lisää riskiä tulla havaituksi (Vacca 2013, 531). Toiminnat voivat tarkoittaa esimerkiksi verkon kartoitusta sekä käyttöjärjestelmän ja käynnissä olevien palvelujen tunnistamista (Melmeg 2014, 19). Taulukossa 1 havainnollistetaan tiedonkeräämisen etenemisen vaiheita.

Taulukko 1. Tiedonkeräämisen eteneminen.

Tiedonkerääminen					
Avoimesti julkistettu tiedustelu		**Ihmisen tiedustelu**	**Jalanjäljet**		**Suojaus-mekanismien kartoitus**
Organisaatio	**Työntekijä**		**Sisäinen**	**Ulkoinen**	
Toiminta-alue	Sosiaalinen Verkosto	Käyttäjän manipulointi (social engineering)	Porttiskannaus	Osoitteet	Toimipisteen tutkiminen
Tuotteet	Blogit	Avaintyönte kijät	Ping/snmp -etsintä	Järjestelm ät	Verkon suojaus
Markkinointi	Internet/Mo biili -jalanjäljet	Partnerit	Aluesiirto	Päivitykse t	Isäntä-koneiden suojaus
Yhteistyökump panit	Historia	Toimittajat	SMTP	Verkkojen kartoitus	Sovellustason suojaus
Avoimet työpaikat	Yhteydet		Forward/Rever se DNS	Haavoittu vat web-sovellukse t	Tallennusjärjestel mien suojaus
Talous			Bannerit		
jne.			VoIP-kartoitus		
			ARP-tutkimus		
			DNS-tutkimus		

Hyökkäysvaihe

Ulkoreunan testaaminen: Tässä vaiheessa tunkeutumistestaaja voi edetä esimerkiksi seuraavanlaisesti: ulkoreunan tunkeutumistestaus (pääseminen palomuurin läpi), osoitettuun kohteeseen murtautuminen ja hallintaan ottaminen, root-oikeuksien käyttäminen ja erilaisten palvelujen asentaminen (Melmeg 2014, 20). Näiden toimintojen jälkeen olisi mahdollista asentaa esimerkiksi rootkit-ohjelmisto tai erinäisiä troijalaisia järjestelmään. Lopuksi suoritetaan mahdollisten jalanjälkien piilottaminen hyökkäyksistä. Tämän vaiheen normaaleja käytäntöjä ovat myös seuraavat asiat:

- Tarkistetaan, miten kohde reagoi murtautumiseen ja miten se pyrkii niitä korjaamaan.
- Testataan kohteen palomuurin reagointia käyttämällä siihen erikseen luotuja paketteja.
- Testataan denial-of-service -hyökkäyksien sietokynnystä lähettämällä muunneltuja TCP- sekä UDP-paketteja.
- Testataan, mitkä protokollasuodattimet ovat käytössä yrittämällä yhdistää kohteeseen yleisimmillä käytetyillä protokollilla, kuten esimerkiksi SSH:lla, FTP:llä ja Telnetillä.
- Testataan hyväksyykö IDS, eli Intrusion Detection System, haitallisen sisällön, ja skannataan kohde monin eri tavoin. Tarkoituksena on testata, kaappaako IDS epänormaalia liikennettä vai ei.
- Testataan, jos järjestelmät demilitarisoidun alueen sisällä vastaavat verkkopalvelinskannauksiin suorittamalla erilaisia menetelmiä kuten POST, DELETE ja COPY. Demilitarisoitu alue voi olla esimerkiksi www-palvelin. (Melmeg 2014, 20.)

Verkkosovellusten testaaminen: Se voi sisältää muun muassa seuraavia toimintoja:

- Testataan, onko kohde altis puskurin ylivuotohyökkäyksille (buffer overflow attack).
- Tarkistetaan, onnistuuko denial-of-service -hyökkäys tai tekemällä liian paljon pyyntöjä palvelulle.
- Testataan, saadaanko arkaluonteista tietoa haltuun välimuistin kautta ja tarkastelemalla virheilmoituksia. (Melmeg 2014, 19.)

Kohteen analysointi: Tässä vaiheessa yritetään löytää kohteesta niin paljon tietoa kuin mahdollista. Tietoja voidaan käyttää myöhemmin, kun hyökkäys tapahtuu. Kohdetietokoneille tehdään erilaisia skannauksia, kuten esimerkiksi haavoittuvuusskannaus ja tunnusteluskannaus (Vacca 2013, 531). Tarkoituksena on testata erilaisia menetelmiä kohteen hakemiselle. Tässä vaiheessa hyökkääjä voi myös yrittää päästä käsiksi kohdetietokoneeseen. Yksi keino on yrittää tehdä käyttäjän manipulointihyökkäyksiä (Melmeg 2014, 19).

Suoritusoikeuksien muuttaminen: Järjestelmän hakemisen jälkeen tavoitteena olisi päästä muuttamaan käyttäjien suoritusoikeuksia hyödyntämällä kohteen tietoja, ja näin ollen päästä käsiksi suojattuihin tiedostoihin. Autentikoidun tilan saavuttamiseksi voidaan käyttää esimerkiksi brute force -hyökkäystä salasanan murtamiseen, troijalaisia tai protokolla-analysaattoreja. (Melmeg 2014, 19.)

Yhteyden luominen: Tässä vaiheessa tunkeutumistestaaja hyödyntää kohteen haavoittuvuutta suorittamalla koodin, joka esimerkiksi mahdollistaisi pääsyn kohteen komentoriville. Yhteyden luomisen jälkeen hyökkääjä voi asentaa esimerkiksi rootkittejä tai implanttiohjelmia, jotka tarjoavat pääsyn järjestelmän sisälle (Melmeg 2014, 20). Yhteyden onnistunut luominen on hyökkääjän kannalta hyödyllistä. Tämän jälkeen hyökkääjän täytyy suojata jälkensä esimerkiksi manipuloimalla lokitietoja. Tämän vaiheen tärkein tavoite on tutkia, mitkä suojausmekanismit mahdollisesti pettävät.

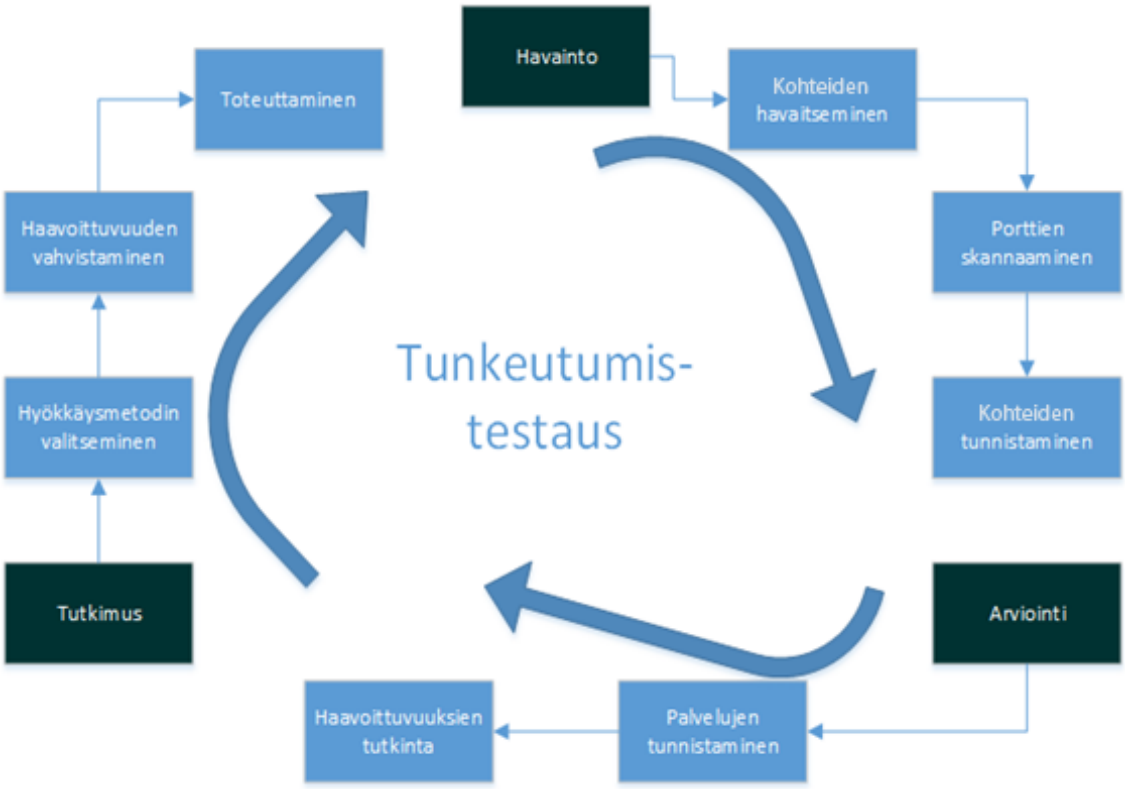
Hyökkäyksen jälkeinen vaihe: Tässä vaiheessa tunkeutumistestaajan täytyy palauttaa järjestelmät takaisin niiden alkuperäisiin tiloihinsa. Tämä sisältää muun muassa toimintoja, joilla poistetaan rootkitit tai muut ohjelmat, jotka luovat niin sanotun takaoven järjestelmään (Vacca 2013, 531). Tietoturvan vuoksi hyödynnetyt haavoittuvuudet on poistettava ja rekisterimerkinnät, joita on tehty haavoittuvuuksien hyödyntämisen ja ohjelmien asentamisen aikana, on puhdistettava. Lisäksi testaajan olisi poistettava yhteydet, joita avattiin yhteyden saamisen aikana (Melmeg 2014, 20).

Tunkeutumistestauksen tuotokset: Tämä vaihe sisältää yksityiskohtaisen raportin kaikista tilanteista ja toiminnoista, joita testauksen aikana on tapahtunut (Vacca 2013, 531). Testauksen havainnoista, tavoitteista ja suositeltavista toimenpiteistä tehdään raportti.

Tunkeutumistestauksen validointi: Tämä on viimeinen vaihe, kun tunkeutumistestaus on suoritettu. Tällöin tuotetaan raportti suojausmekanismien kyvystä suojautua haavoittuvuuksia vastaan. Niiden suojautumiskykyä voidaan

havainnollistaa käyttämällä esimerkiksi arviointiasteikkoa 1-5, jossa 1 tarkoittaa heikkoa ja 5 vahvaa suojautumiskykyä (Vacca 2013, 532). Raportoinnissa kannattaa myös mainita tehtäviä, joita kannattaisi välittömästi suorittaa kohteen tietoturvan parantamiseksi. Lisäksi raportin luojan tulisi kertoa mitä ja kuinka paljon resursseja yrityksen kannattaisi käyttää yrittäessään suojautua haavoittuvuuksilta tulevaisuuden kannalta. Validointiraportti määrittelee myös mitkä tunkeutumiset olivat onnistuneita ja mitkä epäonnistuneita (Melmeg 2014, 67). Suosituksia annetaan turvaamaan niitä osia järjestelmistä, jotka eivät läpäisseet tunkeutumistestausta, tai yltäneet vaaditulle asteikolle sääntöjen tai turvallisuuspolitiikan edellyttämillä tavoilla.

Tunkeutumistestauksen validointi vahvistaa suojaustoimenpiteiden tärkeyttä koko verkkoympäristössä. Suositukset, joita tulisi toteuttaa, sisältyvät raporttiin. Seuraavaksi tehdään analyysi tietoturva-aukoista, jotka näyttävät organisaation sen hetkisen tilanteen verrattuna haluttuun tilanteeseen (Melmeg 2014, 67). Kuva 3 havainnollistaa tunkeutumistestauksen eri vaiheita.

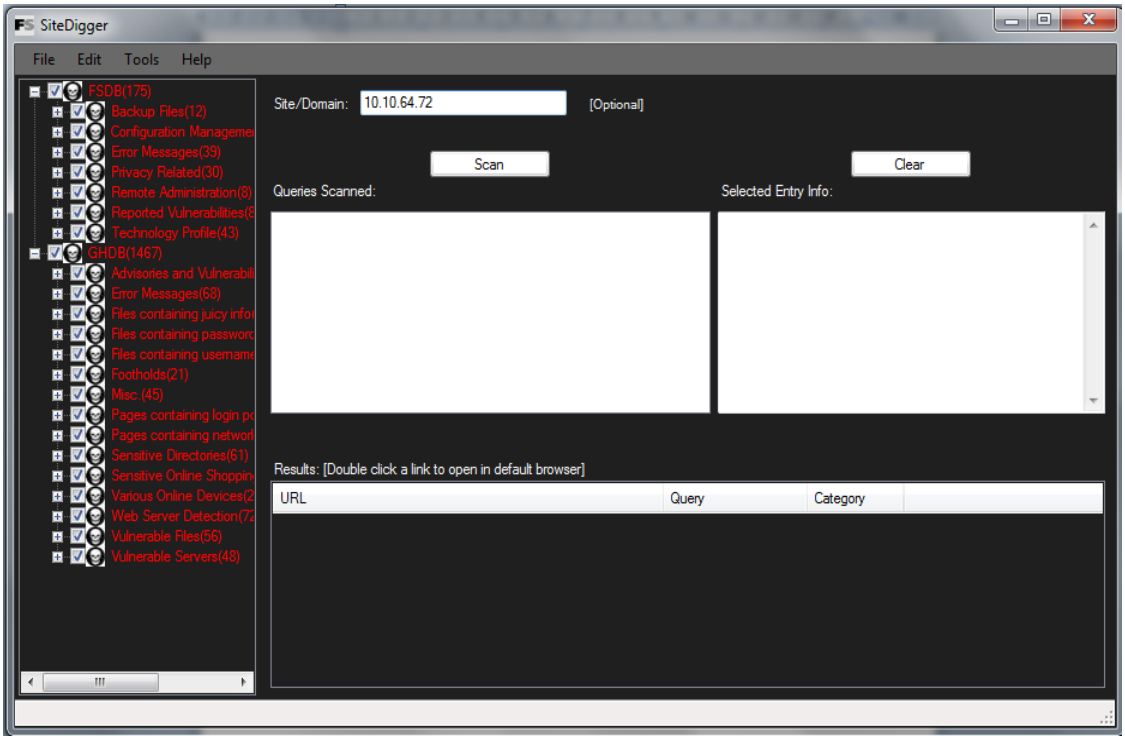


Kuva 3. Tunkeutumistestauksen vaiheet.

4 TIEDONKERUUVAIHEESSA KÄYTETTÄVIÄ SOVELLUKSIA

4.1 SiteDigger

SiteDigger hakee Googlen välimuistista web-sivustojen haavoittuvuuksia, virheitä, asetuksien vuotoja, luottamuksellista tietoa ja tietoturva-aukkoja. Sovelluksessa voi valita tietynlaisia hakuja, joita ohjelma suorittaa. SiteDigger näyttää käyttäjälle, missä muodossa hakulausekkeet suoritetaan. SiteDigger toimii Windows- ja Linux -käyttöjärjestelmillä (McAfee 2014). Kuva 4 havainnollistaa SiteDigger 3.0:n käyttöliittymää.

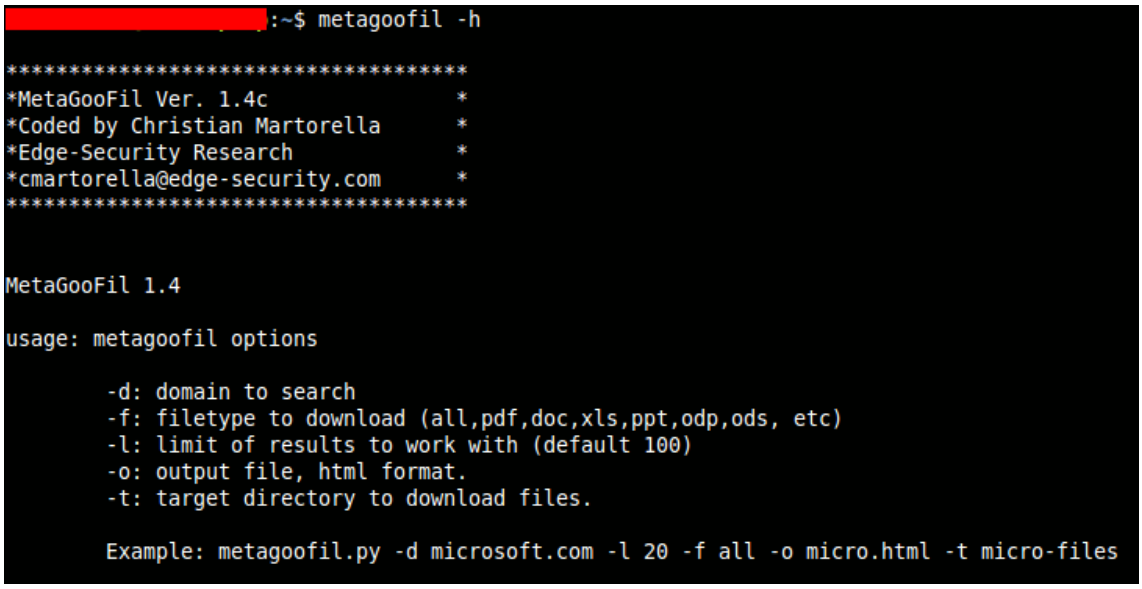


Kuva 4. Kuvakaappaus SiteDigger 3.0:n käyttöliittymästä.

4.2 Metagoofil

Metagoofil on Linux-pohjainen tietojen keräämiseen suunniteltu työkalu. Sen avulla käyttäjä pystyy ottamaan talteen julkisten dokumenttien (.pdf, .doc, .xls,

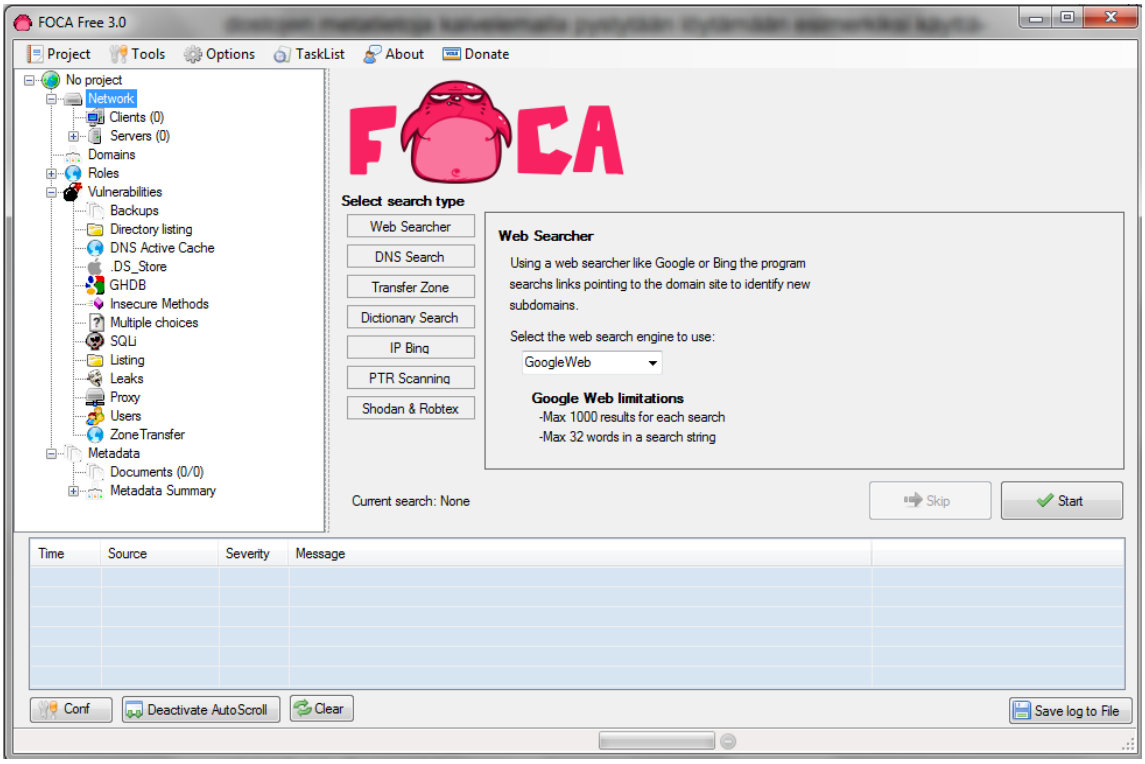
.ppt, .odp, .ods) metadataa talteen saatavilla olevista verkkosivuista. Skannauksen jälkeen Metagoofil luo sivuston, jossa näytetään tulokset metadatan purkamisesta. Lisäksi se luo listan mahdollisista käyttäjätunnuksista, joita voidaan käyttää hyväksi esimerkiksi brute-force -hyökkäyksissä. Lisäksi se purkaa polkuja ja MAC-osoitteita löydetystä metadatasta (Edge-Security 2014). Kuvassa 5 kuvakaappaus Metagoofilin käytöstä Linux-käyttöjärjestelmässä.



Kuva 5. Metagoofilin käyttö Linux-käyttöjärjestelmässä.

4.3 FOCA

FOCA on työkalu, joka lukee asiakirjan ja median metatiedot laajalta alueelta. FOCA hyödyntää Googlen ja Bingin hakukoneita hakemalla verkosta asiakirjoja ja analysoimalla niiden metatietoja. FOCA kykenee löytämään esimerkiksi käyttäjänimiä, polkuja, ohjelmistoversioita, IP-osoitteita, tulostimien tietoja ja sähköpostiosoitteita. Tämän kaiken voi suorittaa lataamatta tiedostoja erikseen (PC & Tech Authority 2014). Kuvassa 6 kuvakaappaus FOCAn käyttöliittymästä.



Kuva 6. Kuvakaappaus FOCAn käyttöliittymästä.

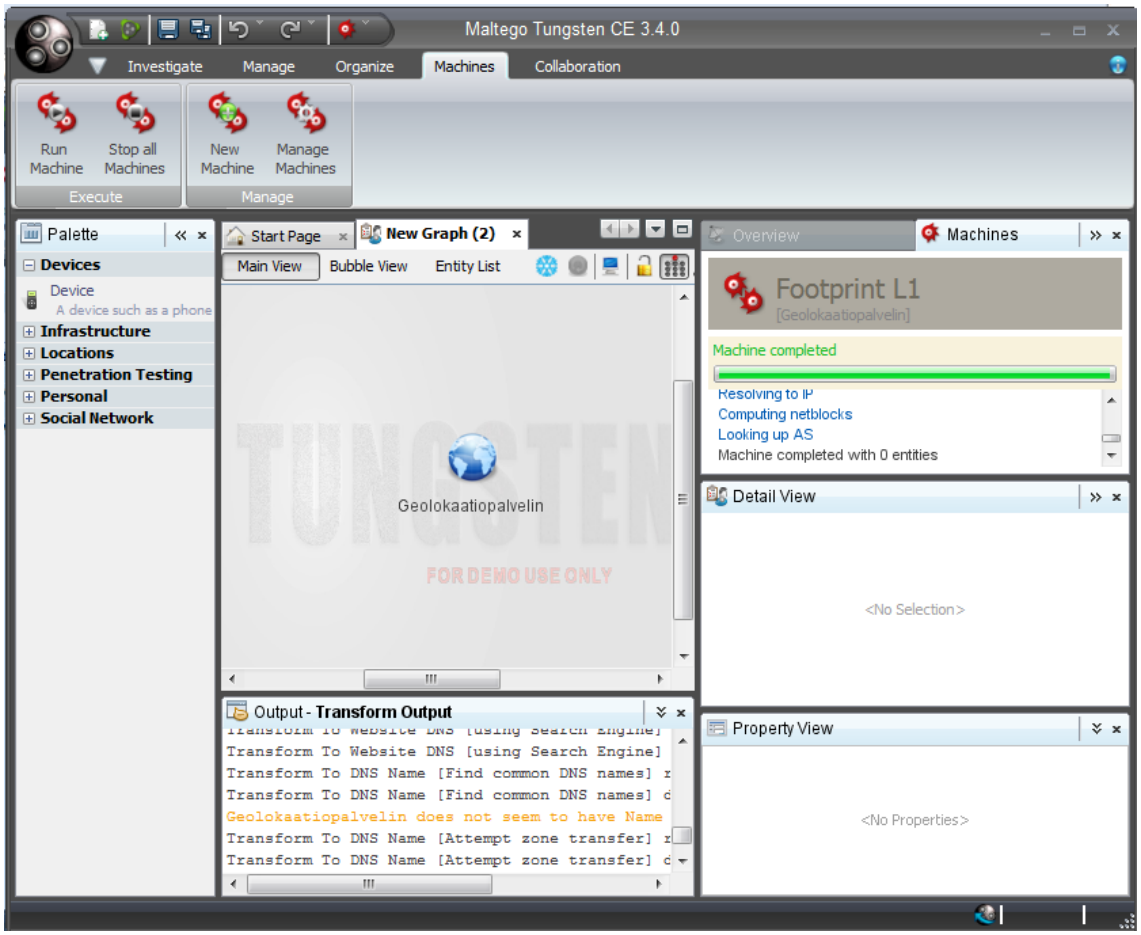
4.4 Maltego

Maltegolla on oma sovellusalusta, joka on kehitetty antamaan selkeän kuvan mahdollisista uhista ympäristössä. Maltego pystyy demonstroimaan haavoittuvuuksien monimutkaisuutta ja vakavuuksia. Maltego on ohjelma, jota voidaan käyttää määrittämään suhteita ja reaalimaailman linkkejä seuraavien asioiden välillä:

1. ihmiset
2. ihmisryhmät (sosiaaliset verkostot)
3. yritykset
4. organisaatiot
5. web-sivustot
6. verkkotunnukset
7. DNS-nimet
8. IP-osoitteet
9. kytkökset

10. asiakirjat ja tiedostot (Paterva 2014).

Kuvassa 7 kuvakaappaus Maltegon käyttöliittymästä Windows 7 -käyttöjärjestelmässä.



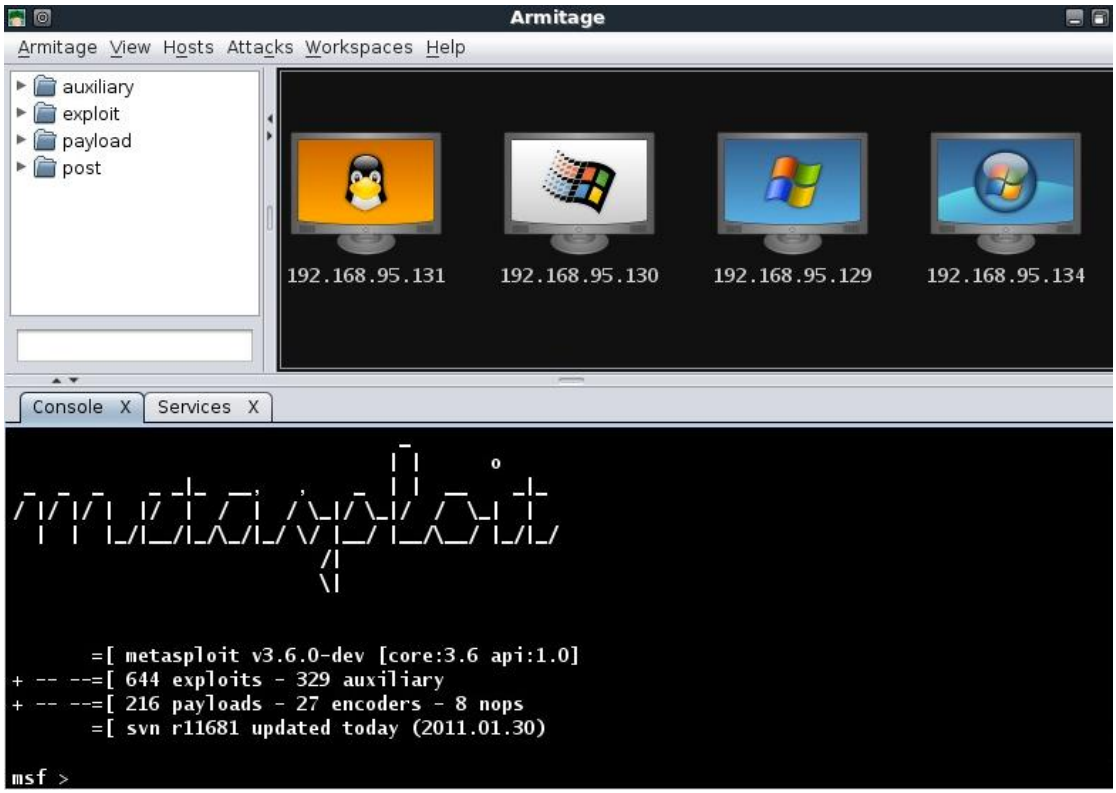
Kuva 7. Kuvakaappaus Maltegon käyttöliittymästä.

5 TIEDON KERÄÄMINEN JA HAAVOITTUVUUKSIEN KARTOITTAMINEN

Tunkeutumistestaamisen edetessä on tärkeää dokumentoida kaikki sen vaiheet. Dokumentointi on tärkeää, jotta havainnot voidaan lopuksi raportoida mahdollisimman tarkasti. Tunkeutumistestaamisen dokumentointiin käytin apuvälineinä kuvakaappauksia, jotka ovat liitettyinä tähän raporttiin. Raporttia voitaisiin käytännössä käyttää loppuraportin osana asiakkaalle.

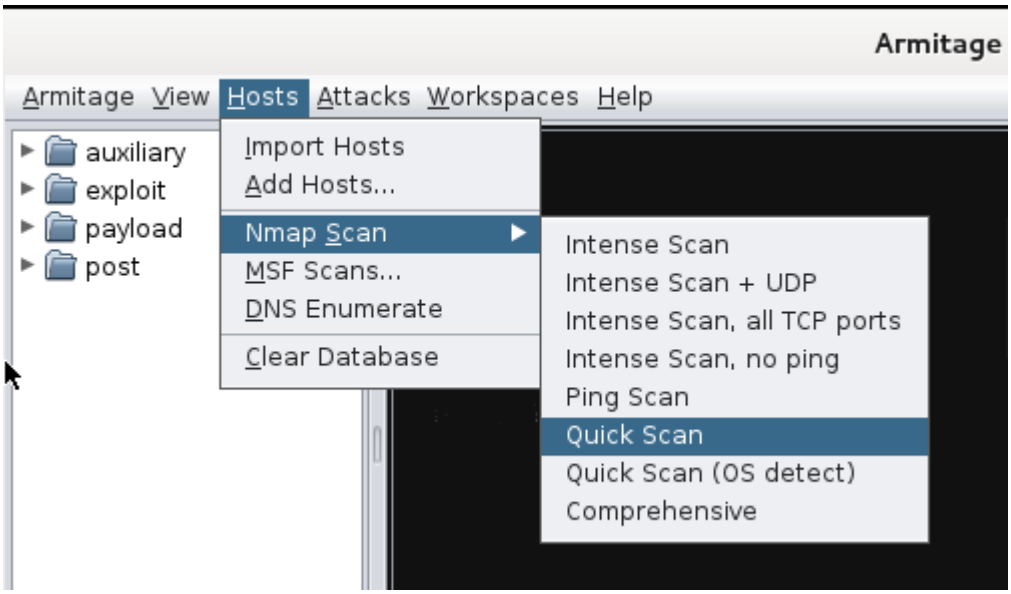
5.1 Armitagen käyttö tiedon keräämiseen

Armitage on graafinen tietoverkkohyökkäyksien hallintatyökalu Metasploitille. Se muun muassa visualisoi kohteet, suosittelee hyödyntämiseen tarkoitettuja työkaluja ja paljastaa lisävaihtoehtoja jälkikäsittelylle (Fast and easy hacking 2014). Kuvassa 8 kuvakaappaus Armitagen käyttöliittymästä.



Kuva 8. Armitagen käyttöliittymä.

Testauksessa suoritetaan Armitageen integroidulla työkalulla Nmap (Network Mapper) skannaus tietokoneiden etsimiseksi. Nmap on turvaskanneri, jota käytetään tietokoneiden ja palveluiden etsimiseen ja kartoittamiseen verkossa. Siitä tulee nimitys ”map” eli kartta. Saavuttaakseen tavoitteensa Nmap lähettää tietyllä tavalla muodostettuja paketteja kohdekoneisiin, minkä jälkeen se analysoi niiden antamia vastauksia. Kuvassa 9 Armitagen sisältämän Nmap-skannaustyökalun valitseminen.



Kuva 9. Armitagen sisältämä Nmap-skannaustyökalu.

Aluksi ohjelmassa valitaan IP-alue, josta palvelimen IP-osoite halutaan etsiä. Tavoitteena on löytää geolokaatiopalvelimen osoite. Otetaan DHCP pois käytöstä.

Skannauksen jälkeen Armitage näyttää löydetyt tietokoneet graafisesti. Skannaus löysi tavoitellun tietokoneen. Kuvassa 10 on esitetty tuloste Nmapin skannauksesta ja sen löytämistä tietokoneista.

Kuva 10. Nmapin löytämiä isäntiä Armitage-ohjelmassa.

Testauksessa yritetään etsiä avoimia portteja Nmap-työkalun avulla. Komento nmap -SV [ip-osoite] etsii kohdekoneesta 1024 porttia. Tämän jälkeen se listaa avoimet portit ja niissä olevat palvelut. Nmap-skannaus suoritettiin myös manuaalisesti. Kohdekoneen IP-osoite oli väliaikaisesti vaihtunut osoitteesta 10.10.64.72 osoitteeseen 10.10.64.105, koska DHCP, eli Dynamic Host Configuration Protocol, oli käytössä. DHCP:n tehtävänä on jakaa IP-osoitteita uusille lähiverkkoon kytkeytyville laitteille. Kuvassa 11 on kuvakaappaus suoritetusta Nmap-skannauksesta.

```
root@kalilinux:~# nmap -sV 10.10.64.105

Starting Nmap 6.40 ( http://nmap.org ) at 2014-04-01 12:53 BST
Nmap scan report for 10.10.64.105
Host is up (0.00081s latency).
Not shown: 996 filtered ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 6.0p1 Debian 4 (protocol 2.0)
80/tcp  open  http     Apache httpd 2.2.22 ((Debian))
111/tcp open  rpcbind  2-4 (RPC #100000)
443/tcp open  ssl/http Apache httpd 2.2.22 ((Debian))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at http://nmap.
org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.68 seconds
root@kalilinux:~#
```

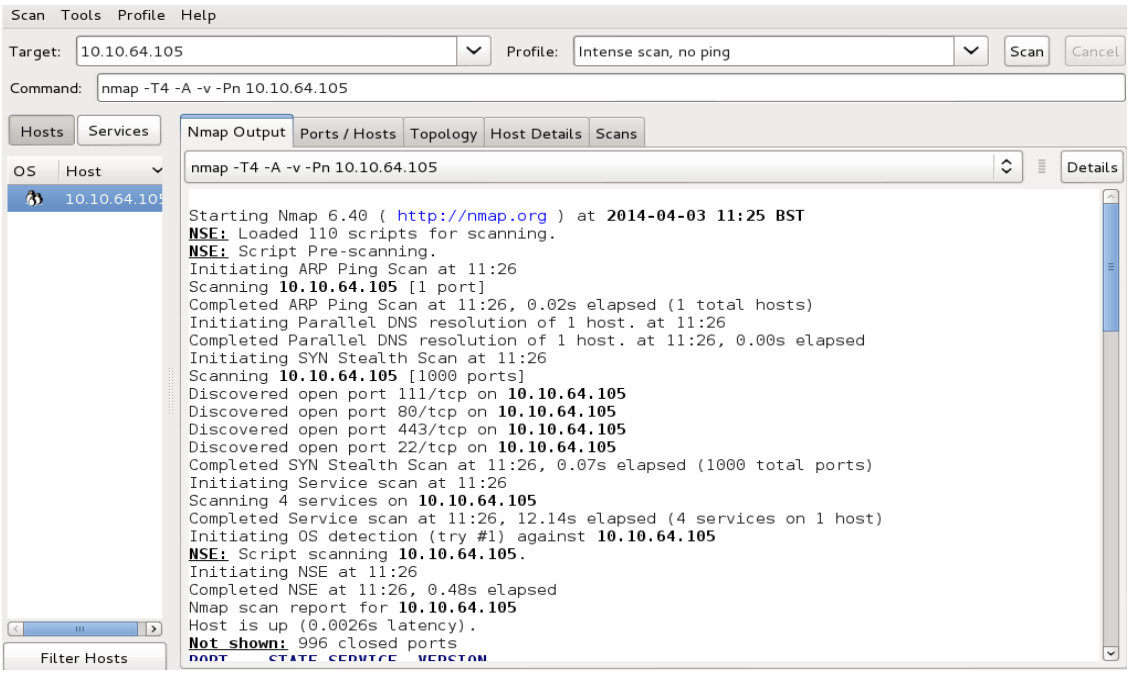
Kuva 11. Nmap-skannaus kohdekoneen 1024:sta portista.

5.2 Zenmapin käyttö tiedon keräämiseen

Seuraavaksi tutkitaan, miten geolokaatiopalvelin vastaa Zenmap-sovelluksella tehtyyn porttiskannaukseen. Suoritetaan haku komennolla nmap *-T4 -A -v -Pn 10.10.64.105*.

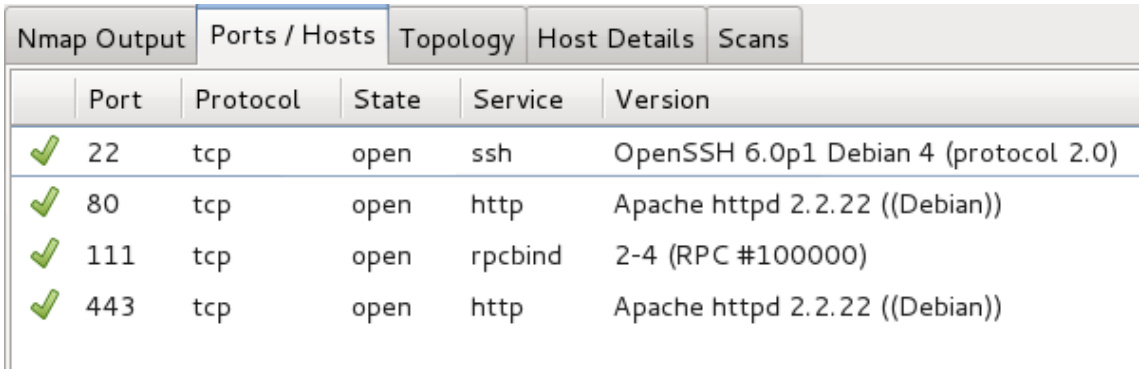
- T4 = asetetaan skannausnopeudeksi 4 (asteikolla 0-5, mitä suurempi luku - sitä nopeampi skannausnopeus). Mitä nopeampi skannaus on, sitä suurempi riski on joutua huomatuksi. Tässä tapauksessa nopeudella ei kuitenkaan ole väliä.
- A = ottaa käyttöön käyttöjärjestelmän, version ja ohjelmakoodin havaitsemisen. Lisäksi traceroute otetaan käyttöön. Traceroute on TCP/IP-protokollaa käyttävä työkalu, joka selvittää, mitä reittiä protokollan paketit siirtyvät määrättyyn koneeseen.
- v = tulosteen tarkkuus, määrittelee paljonko skannaus antaa suorittaessaan tietoa (mitä enemmän v-kirjaimia, sitä monisanaisempi haku, esimerkiksi -vvvv olisi taso 4).
- Pn = ottaa yhteyskokeilun pois käytöstä, koska oletetaan, että yhteys onnistuu. (Linuxmanpages 2014.)

Kuvakaappaus Zenmapin käyttöliittymästä ja skannauksesta näkyy kuvassa 12.

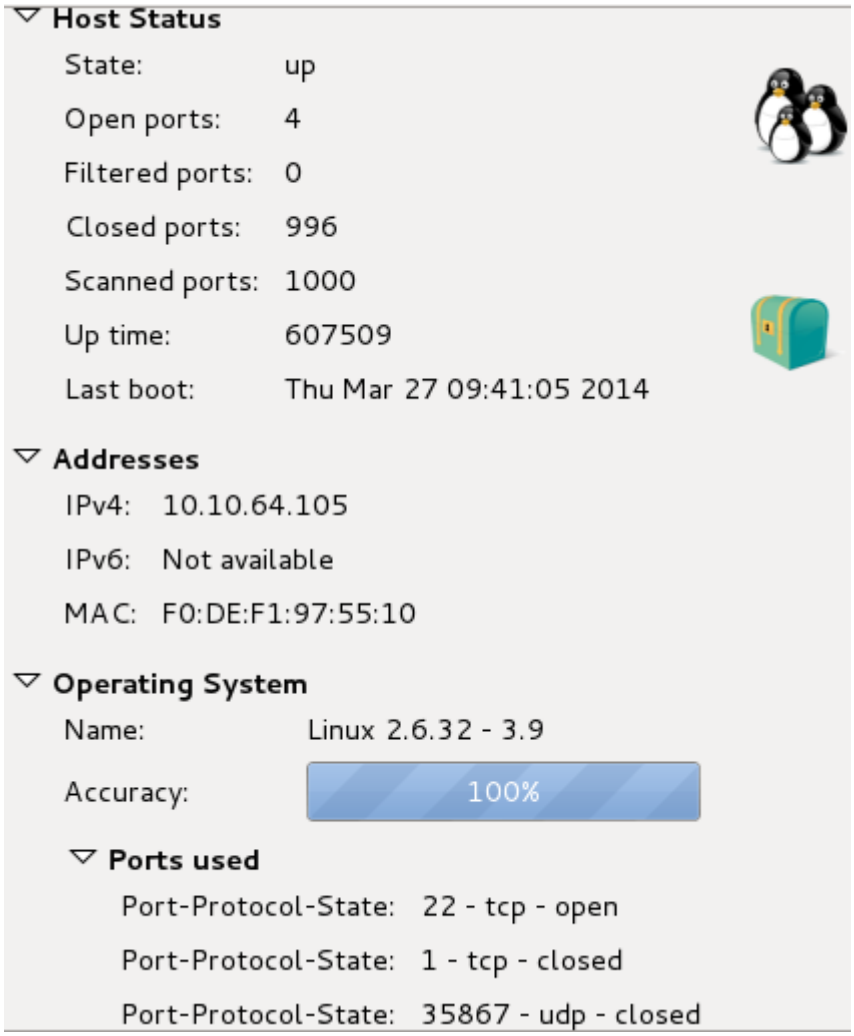


Kuva 12. Skannaus kohdekoneesta Zenmap-sovelluksella.

Skannauksen jälkeen Zenmap kertoo tietoja kohdekoneesta, kuten esimerkiksi käyttöjärjestelmästä, IP-osoitteesta, MAC-osoitteesta ja auki olevista porteista. Skannatuista 1000:sta portista sovellus löysi neljä avointa ja 996 suljettua porttia. Avoimet TCP-portit ovat 22, 80, 111 ja 443. Avoimissa porteissa suoriutuvat palvelut ovat ssh, http ja rpcbind. Kohdekoneen käyttämä käyttöjärjestelmä on Linux. Kuvassa 13 on esitetty kuvakaappaus avoimista porteista. Kuvassa 14 on Zenmapin luoma havainnollistus skannauksen löytämistä tuloksista.



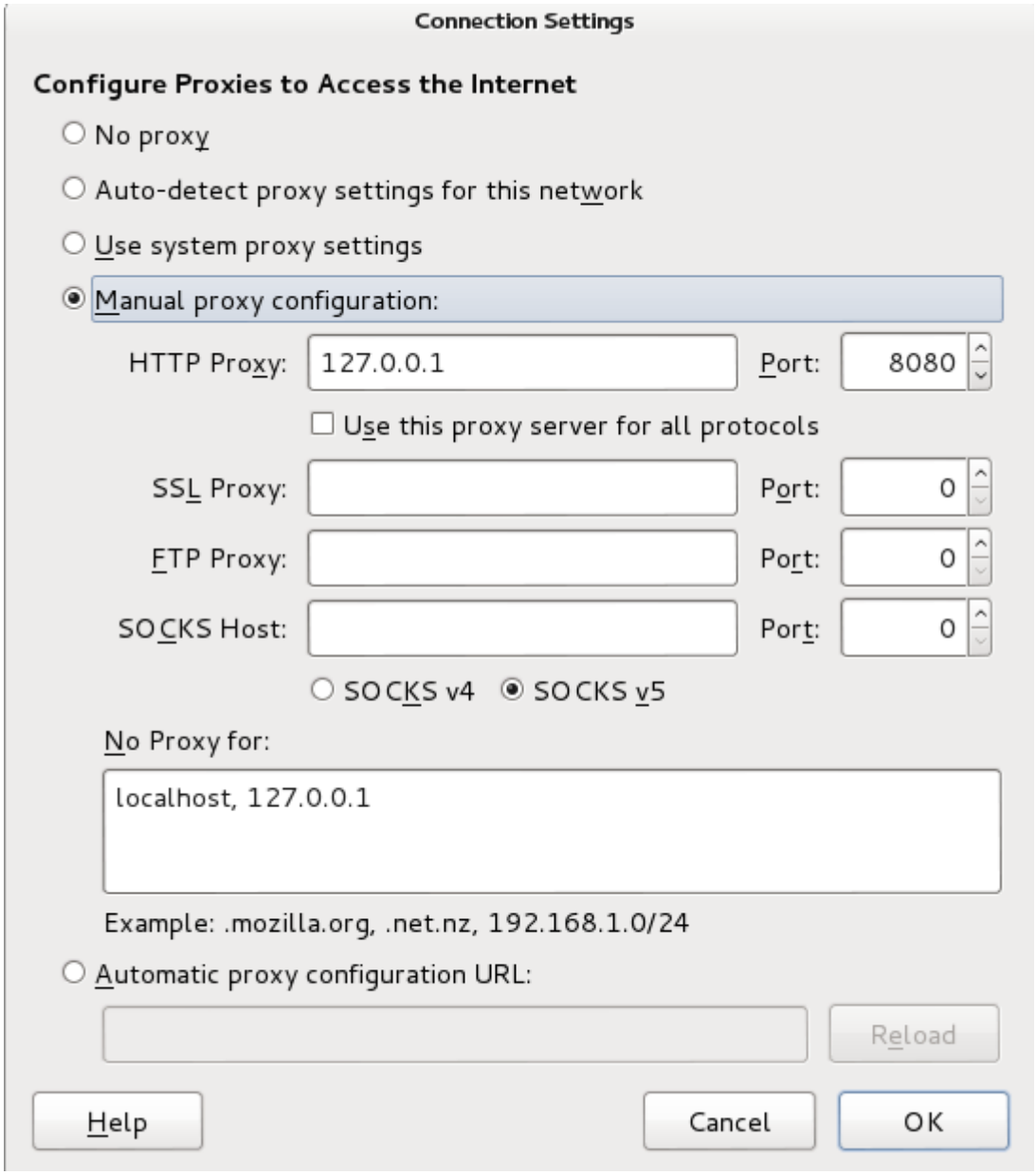
Kuva 13. Zenmapin löytämät avoimet portit ja niistä löytyvät palvelut.



Kuva 14. Zenmapin skannauksen tuloksia.

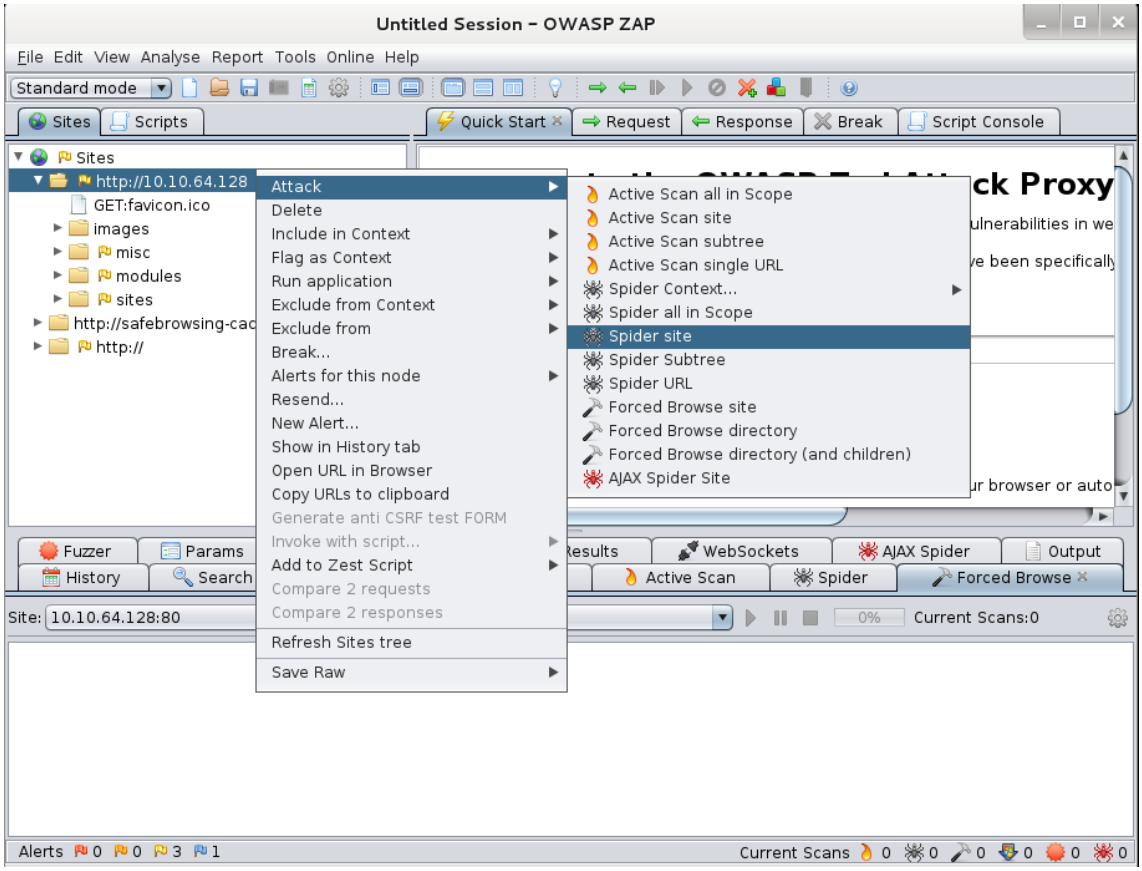
5.3 Web-sovelluksien testaaminen

Kuvassa 15 on web-sovellusten testaamista varten asetetut Firefox-selaimen proxy-asetukset OWASP ZAP -sovellusta varten.



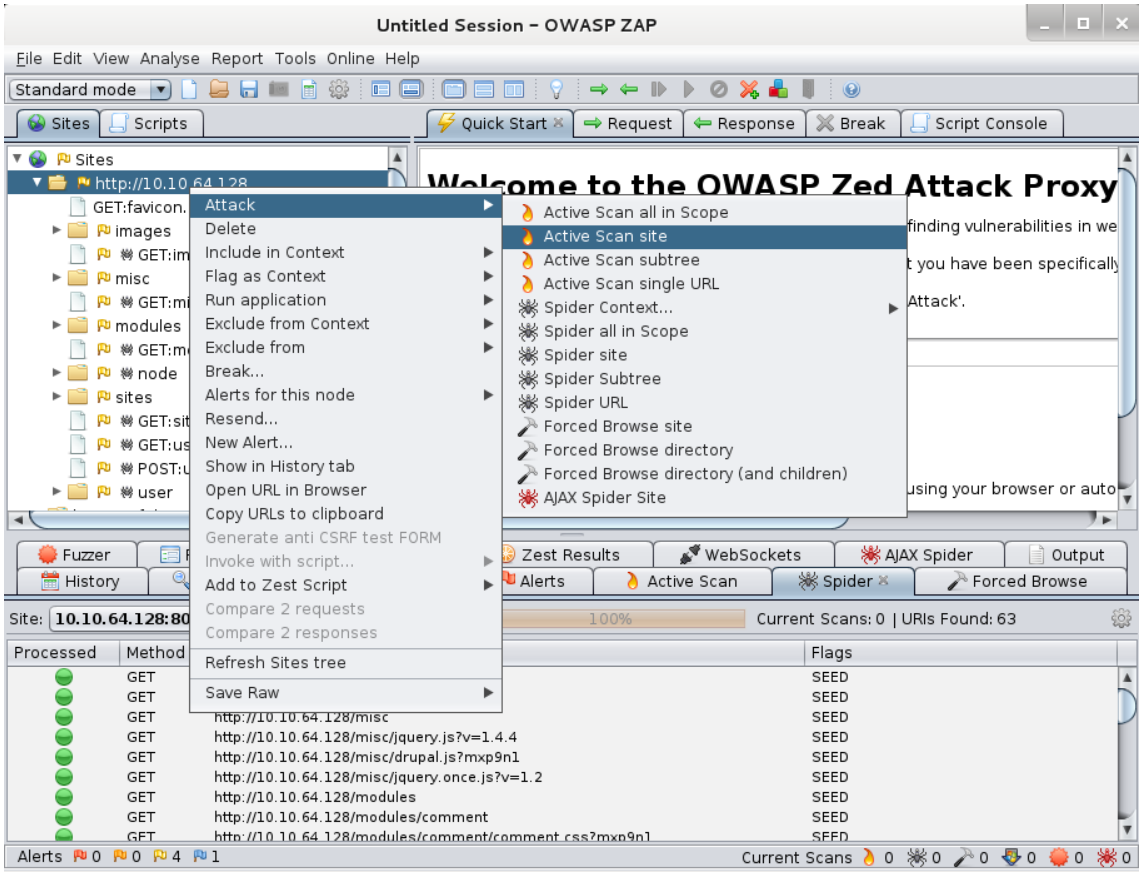
Kuva 15. Firefox-selaimen proxy-asetukset.

Käytetään geolokaatiopalvelimen kartoittamiseen OWASP ZAP -sovelluksen sisältämää Spider site -työkalua. Kuvassa 16 on Spider site -työkalun valitseminen.



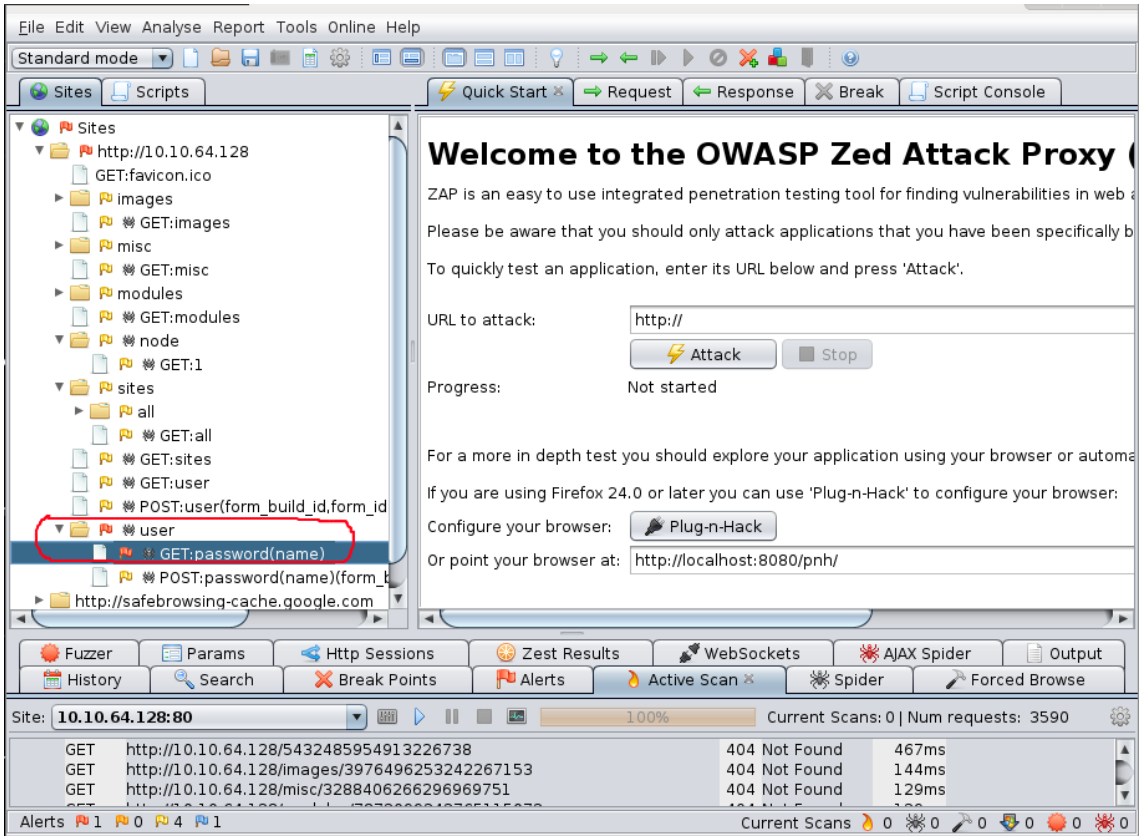
Kuva 16. Spider site -työkalun valitseminen.

Käytetään Active Scan site -työkalua spider siten löytämien sivujen haavoittuvuuksien testaamiseen. Kuvassa 17 on Active Scan site -työkalun valitseminen.



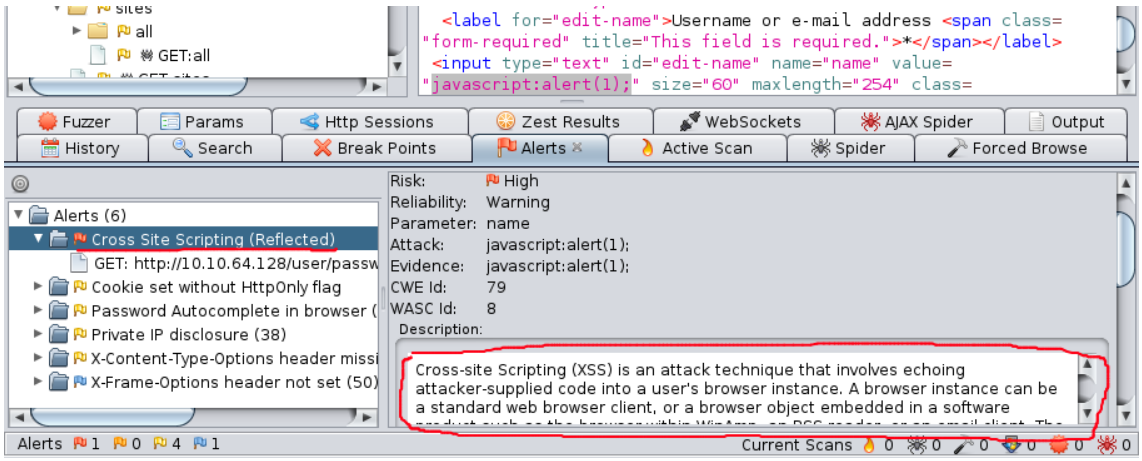
Kuva 17. Active Scan site -työkalun valitseminen.

Active Scan site -työkalu löysi geolokaatiopalvelimelta vain yhden vakavaksi riskiksi luokitellun haavoittuvuuden. Kuvassa 18 Active Scan site -työkalun skannauksen löytämät tulokset. Punaiset liput merkitsevät vakavaksi riskiksi määriteltyä haavoittuvuutta.



Kuva 18. Skannauksen löytämät vakavan luokan haavoittuvuudet.

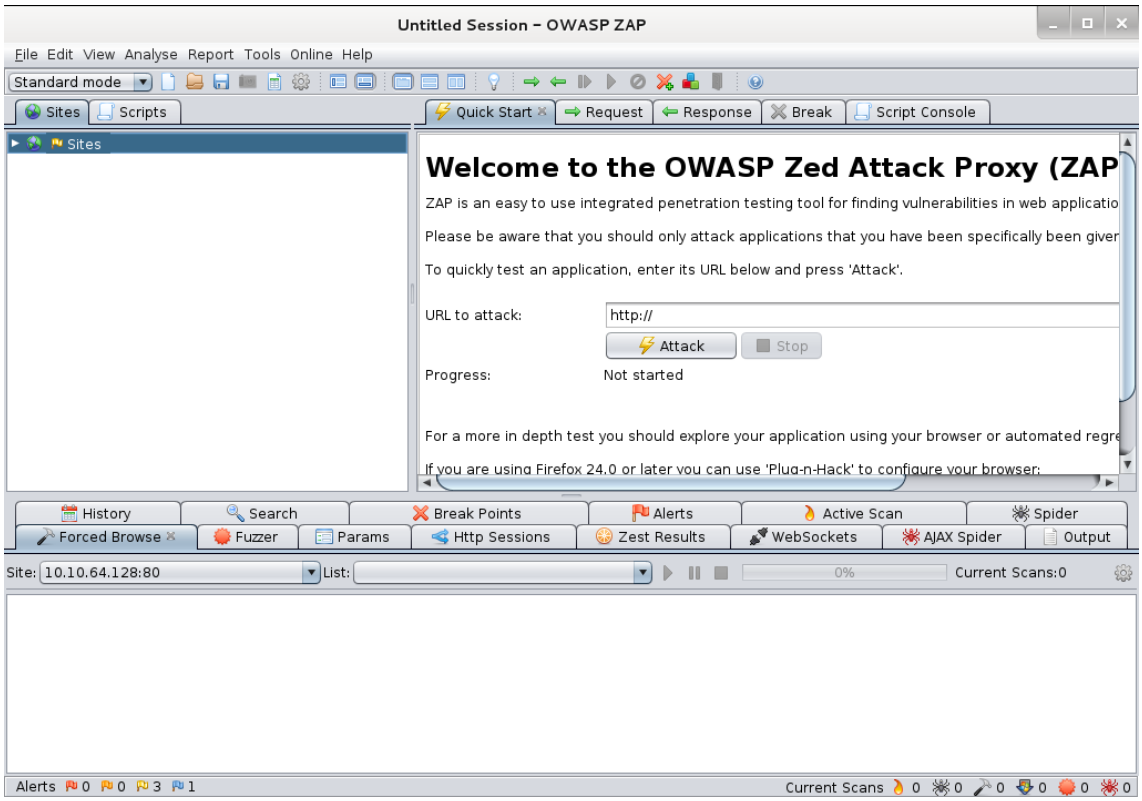
Skannaus ilmoittaa sivuston olevan haavoittuvainen XSS-injektio-hyökkäyksille. Kuvassa 19 on havainnollistus skannauksen ilmoittamasta haavoittuvuudesta.



Kuva 19. Sivusto on haavoittuvainen XSS-injektiohyökkäyksille.

5.4 OWASP ZAP

Zed Attack Proxy (ZAP) on integroitu tunkeutumistestaamiseen tarkoitettu työkalu web-sovellusten haavoittuvuuksien löytämiseen. ZAP sisältää automatisoituja skannereita ja joukon työkaluja, joiden avulla voi löytää tietoturva-aukkoja manuaalisesti. ZAP on kehittynyt apuohjelma, jonka avulla voi suorittaa tunkeutumistestaamista tietoturvariskien sekä verkkosovellusten heikkojen kohtien löytämiseen. Kuvassa 20 kuvakaappaus OWASP ZAPin käyttöliittymästä Linux-käyttöjärjestelmässä. (OWASP 2014)

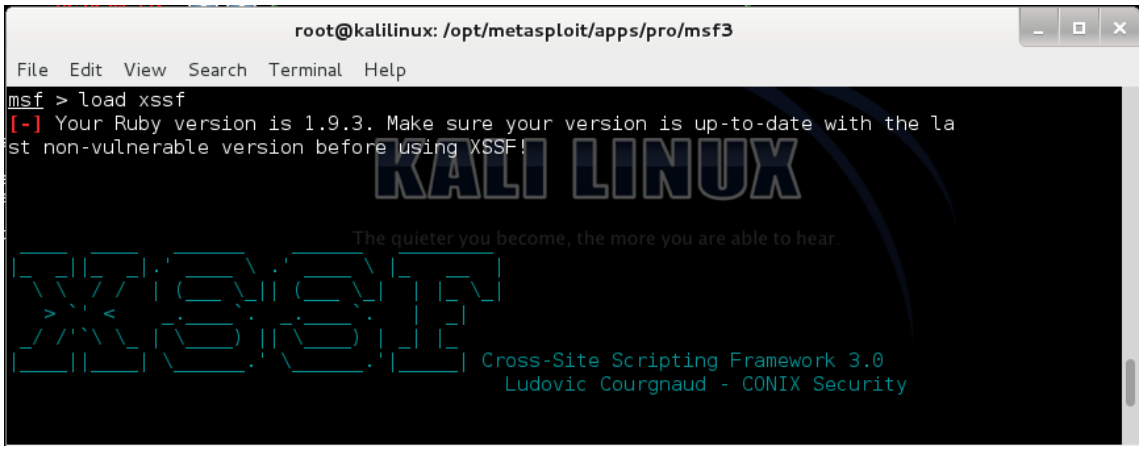


Kuva 20. Kuvakaappaus OWASP ZAPin käyttöliittymästä.

5.5 XSSF

Cross-Site Scripting Framework (XSSF) on turvallisuustyökalu, joka on suunniteltu tekemään XSS-haavoittuvuuden hyödyntämisestä helpompaa. XSSF-projektin tavoitteena on demonstroida todellisten XSS-haavoittuvuuksien

vaaroista. XSSF voidaan asentaa Metasploit-ohjelmistoon, jonka avulla käyttäjä kykenee tekemään Metasploit Framework -pohjaisia haavoittuvuuksien hyväksikäyttöjä XSS-haavoittuvuuksille (XSSF 2014). Metasploit Framework on työkalu, jolla voidaan luoda ja suorittaa koodia kohdeverkkoa tai -järjestelmää vastaan. Se sisältää valmiiksi hyökkäyksiin käytettäviä hyödyntämiseen tarkoitettuja työkaluja. Kuvassa 21 on esitetty XSSF:n käyttöä Metasploit-ohjelmistossa.



Kuva 21. XSSF:n käyttö Metasploit-ohjelmistossa.

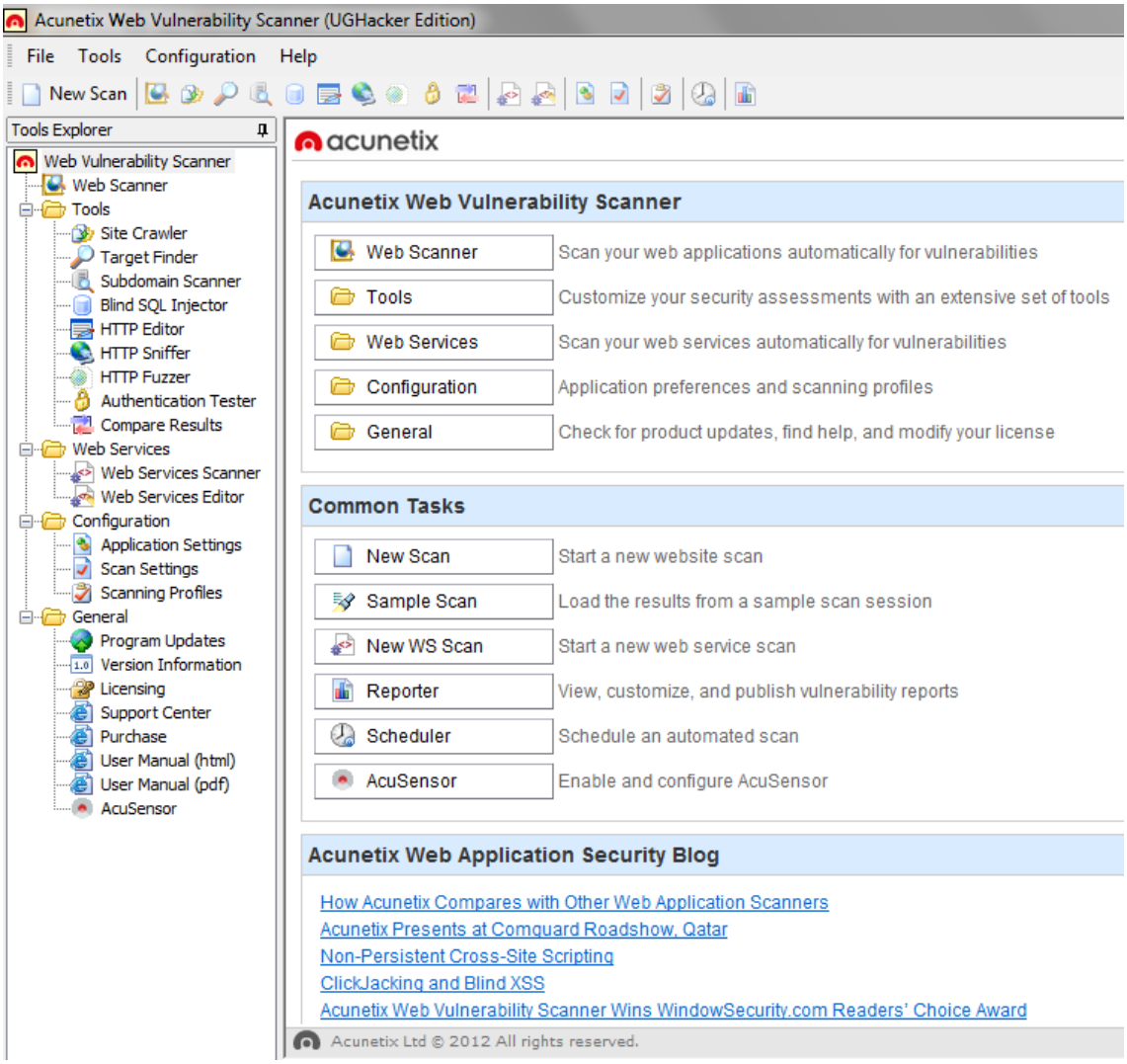
6 HAAVOITTUVUUSSKANNAUSTEN TULOKSET

Seuraavaksi jatketaan tunkeutumistestausta skannaamalla geolokaatiopalvelinta erilaisilla haavoittuvuuksia tutkivilla skannereilla.

6.1 Acunetix Web Vulnerability Scanner

Acunetix Web Vulnerability Scanner on automatisoitu web-sovellusten turvallisuuteen kehitetty testaustyökalu, joka tarkistaa hyödynnettävissä olevia haavoittuvuuksia. Automatisoituja skannauksia voidaan täydentää tai tarkistaa monilla erilaisilla välineillä, jotka mahdollistavat kokonaisvaltaisen web-sovellusten ja web-sivustojen tunkeutumistestaamisen. (WineHQ 2014.)

Hakkerit keskittyvät usein sellaisiin Web-pohjaisiin sovelluksiin, kuten esimerkiksi verkkokaupat, lomakkeet, sisäänkirjautumissivut, dynaamiset sivut ja sivustot, joista voidaan poimia henkilökohtaista tietoa (Acunetix 2014b). Eräs hyvä esimerkki on yhtiöiden tietojen varkaudet, kuten esimerkiksi luottokorttien tiedot ja asiakaslistat. Palomuurit, SSH ja SSL eivät usein riitä, jos haluaa olla turvassa Web-sovellusten hakkeroinnilta. Kuvassa 22 on kuvakaappaus Acunetixin käyttöliittymästä Windows 7 -käyttöjärjestelmässä.



Kuva 22. Acunetixin käyttöliittymä.

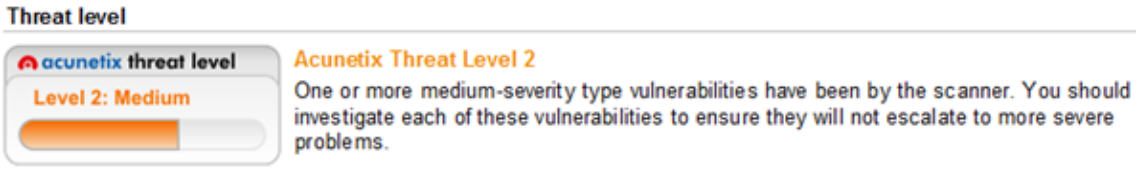
Tulokset

Vakavaksi riskiksi luokitellut haavoittuvuudet ovat kaikista vaarallisimpia. Nämä varoitukset ilmoittavat sivuston olevan hyvin altis hakkereille ja tiedonkaappaukselle.

Keskinkertaisiksi haavoittuvuuksiksi luokitellaan väärin asetetut asetukset ja sivustokoodin virheet. Nämä voivat johtaa esimerkiksi palvelimen häiritsemiseen tai tunkeutumiseen.

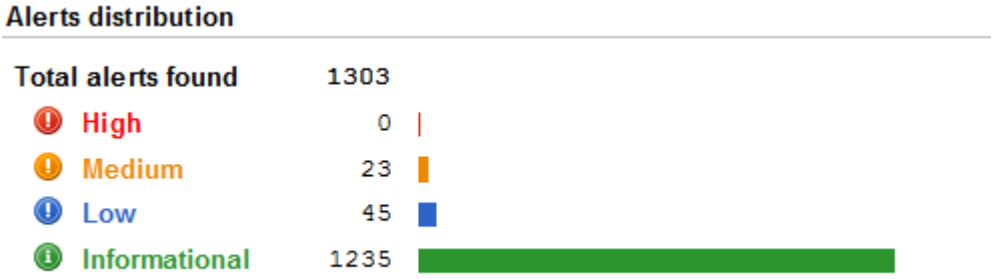
Matalaksi riskeiksi luokitellut haavoittuvuudet ovat muun muassa tietoliikenteen salauksien vähäisyys ja kansioiden sijainnin paljastuminen.

Tietovaroitus ilmaisee asioista, jotka voivat paljastaa tietoja esimerkiksi Google-hakukoneen merkkijonojen käyttämisestä hakkerointimielessä tai sähköpostin reitin paljastamiseen (Acunetix 2013). Seuraavaksi tehdään haavoittuvuuskannaus geolokaatiopalvelimelle käyttäen ohjelmaa Acunetix Web Vulnerability Scanner. Kuvassa 23 kerrotaan uhkatasosta, jonka ohjelma on arvioinut geolokaatiopalvelimesta.



Kuva 23. Acunetixin arvioima uhkataso geolokaatiopalvelimella.

Skannaus löysi 0 vakavan luokan, 23 keskitason luokan, 45 alhaisen luokan ja 1235 informatiivisen luokan haavoittuvuutta. Kuvassa 24 on havainnollistus löydetyistä haavoittuvuuksista.



Kuva 24. Haavoittuvuustasojen jakautuminen geolokaatiopalvelimella.

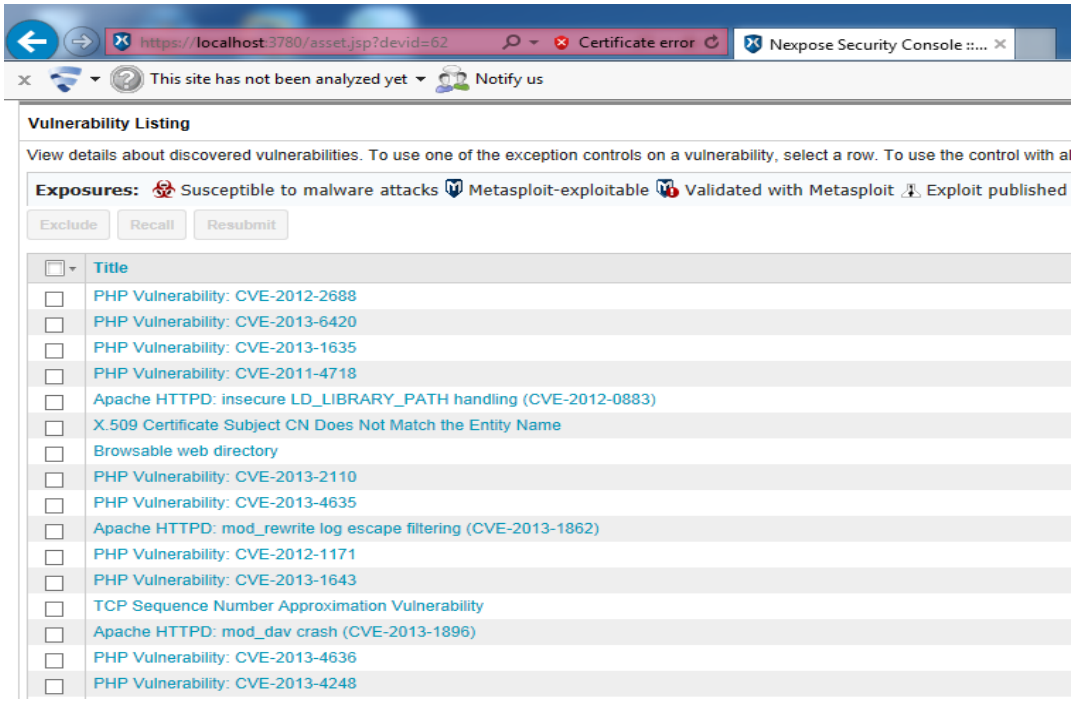
Tiedostojen tarkenteet voivat tarjota informaatiota siitä, mitä teknologiaa verkkosivustolla käytetään. Taulukossa 2 on havainnollistettu skannauksen löytämiä tiedostojen tarkenteita.

Taulukko 2. Tiedostojen tarkenteet.

Tiedostojen tarkenteet	
css	36
js	22
txt	10
rb	1
php	4
html	1641
sty	1
dtd	5
ent	1
properties	1

6.2 Rapid7 Nexpose

Rapid7 Nexpose on haavoittuvuusskanneri, jonka tavoitteena on tukea koko haavoittuvuuden hallinnan elinkaarta. Se sisältää haavoittuvuuksien havaitsemisen, tarkastamisen, riskien luokittelun, vaikutusten arvioinnin ja raportoinnin. Se integroituu Rapid7 Metasploitin kanssa haavoittuvuuksien hyödyntämiseen (Sectools 2014). Kuvassa 25 on kuvakaappaus Nexposen käytöstä selaimella.



Kuva 25. Nexposen käyttö selaimella.

Tulokset

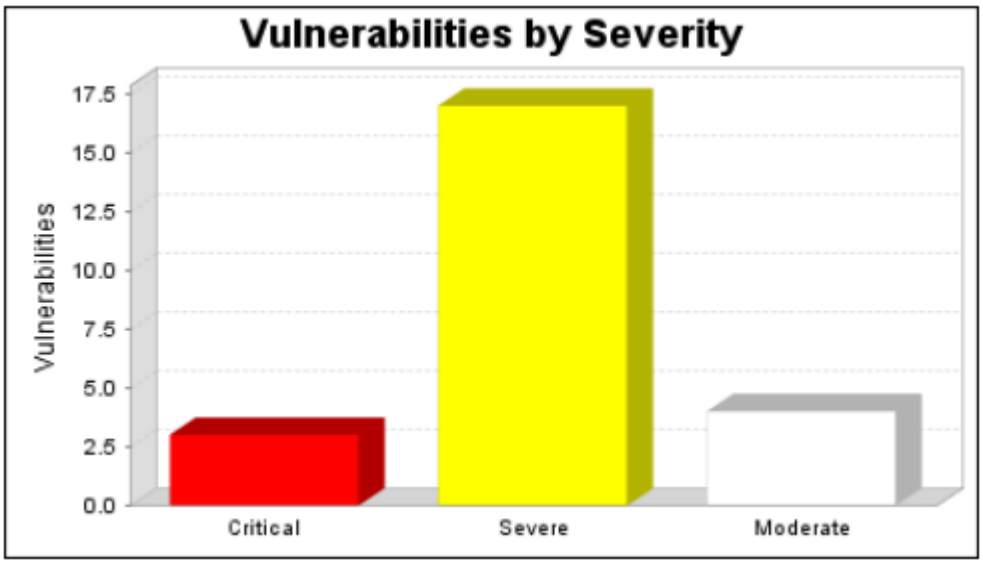
Nexpose Rapid7 LLC -työkalulla tehty skannaus löysi 24 haavoittuvuutta. Näistä kolme oli kriittisiä haavoittuvuuksia. Hyökkääjän on suhteellisen helppo käyttää hyväkseen kriittisiä haavoittuvuuksia. Ne voivat tarjota hyökkääjälle täydet oikeudet järjestelmiin. Kuvassa 26 on kuvakaappaus Nexposen ilmoittamista kriittisistä haavoittuvuuksista.

Title	CVSS	Risk	Published On	Severity
PHP Vulnerability: CVE-2012-2688	10	716	Fri Jul 20 2012	Critical
PHP Vulnerability: CVE-2013-6420	7.5	542	Mon Dec 16 2013	Critical
PHP Vulnerability: CVE-2013-1635	7.5	571	Wed Mar 06 2013	Critical

Kuva 26. Nexposen löytämät kriittiset haavoittuvuudet.

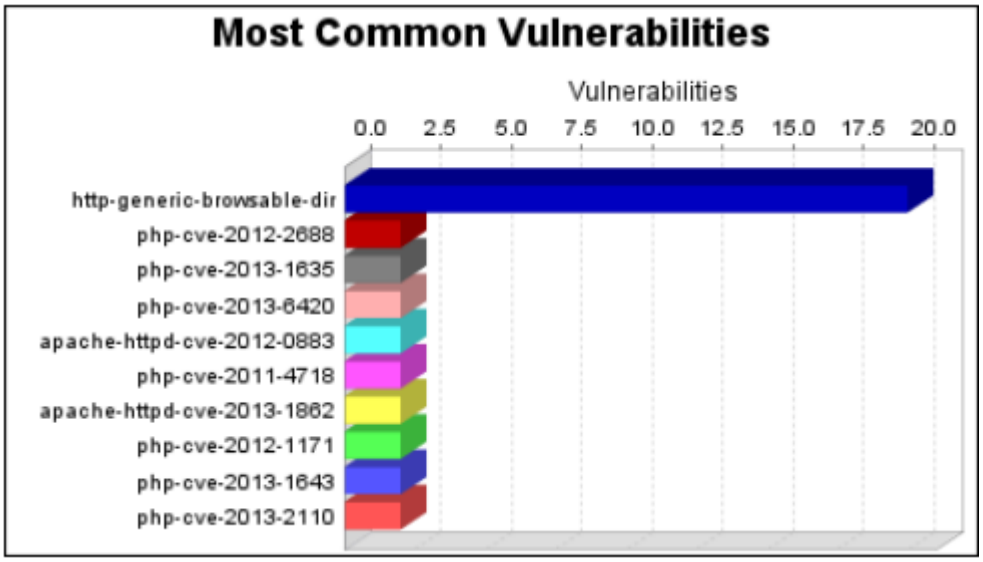
Haavoittuvuuksista 17 oli vakavia. Vakavia haavoittuvuuksia on usein vaikeampi hyödyntää, ja ne eivät välttämättä takaa samoja oikeuksia hyökkääjälle kuin kriittiset haavoittuvuudet.

Haavoittuvuuksista neljä oli keskinkertaisia. Keskinkertaiset haavoittuvudet tarjoavat hyökkääjälle informaatiota, joka auttaa muiden hyökkäyksien toteuttamista. Kuvassa 27 on havainnollistettu haavoittuvuusskannauksen tulokset skannerin löytämistä vakavuuksista geolokaatiopalvelimella.

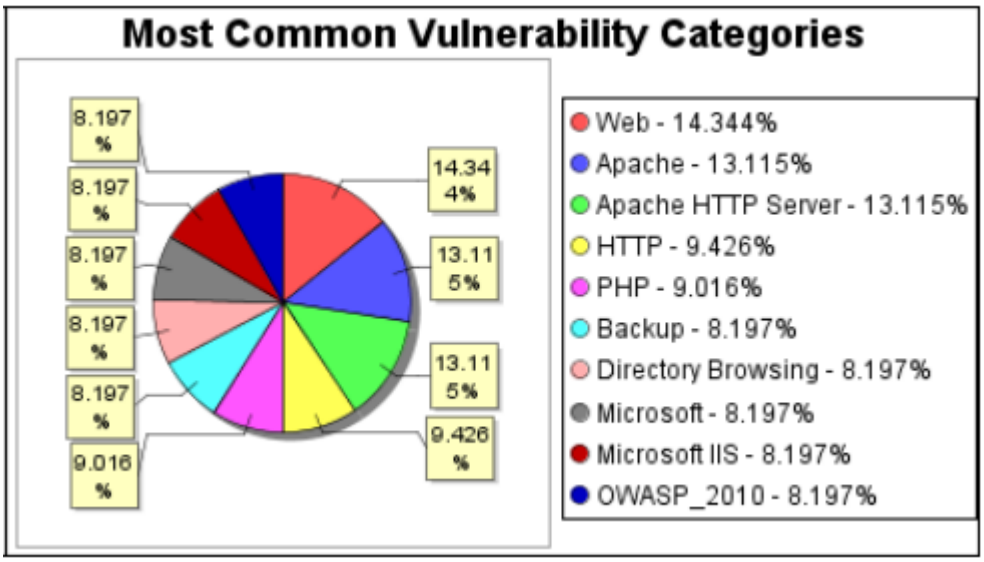


Kuva 27. Haavoittuvuuden vakavuudet geolokaatiopalvelimella.

Skannaus löysi 20 haavoittuvuutta web-kansiosta http-generic-browsale-dir. Täten se oli haavoittuvuuksista yleisin. Web-kategoriasta löytyi 35 haavoittuvuutta. Täten se oli haavoittuvuuskategorioista yleisin. Kuvassa 28 on havainnollistettu yleisimmät haavoittuvuudet. Kuvassa 29 on havainnollistettu yleisimmät haavoittuvuudet luokittain.

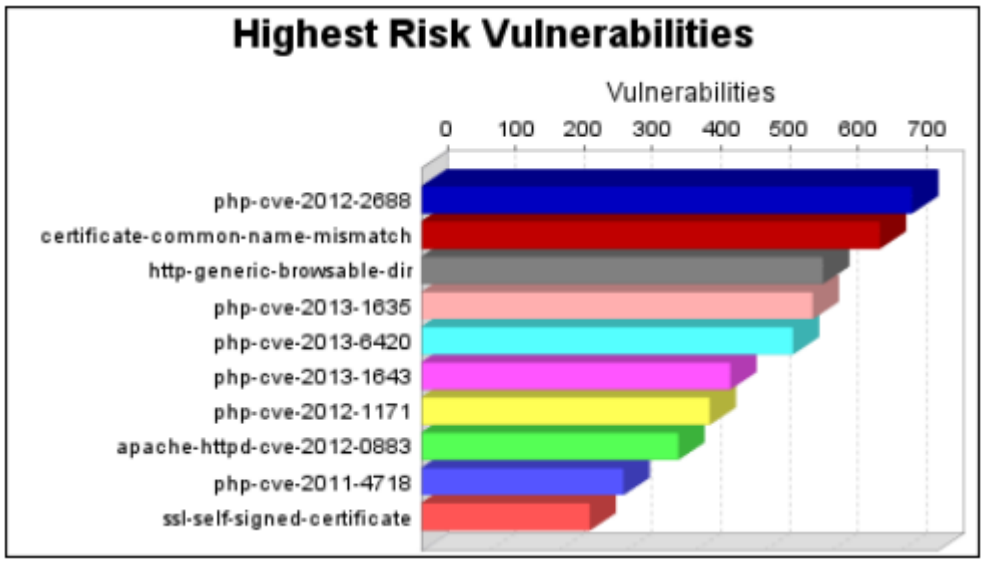


Kuva 28. Yleisimmät haavoittuvuudet.



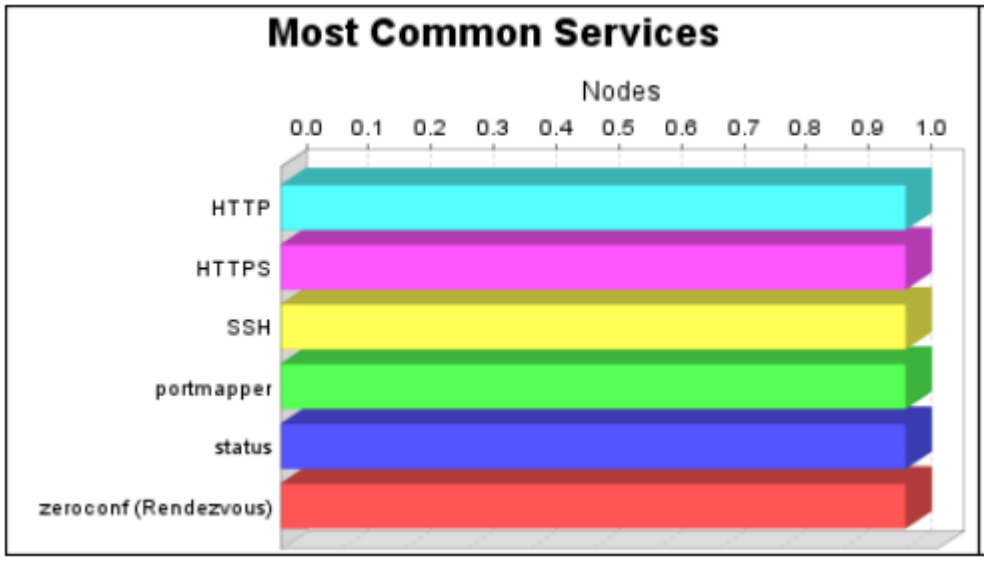
Kuva 29. Yleisimmät haavoittuvuudet luokittain.

Skannaus löysi korkean riskin haavoittuvuuksia. Ohjelma luokitteli näistä haavoittuvuuksista php-cve-2012-2688:n korkeimmaksi riskiksi organisaatiolle. Sen tulos on 716. Kuvassa 30 on havainnollistettu korkean riskien haavoittuvuuksista geolokaatiopalvelimella.



Kuva 30. Korkean riskin haavoittuvuudet geolokaatiopalvelimella.

Skannaus havaitsi 6 geolokaatiopalvelimella käynnissä olevaa palvelua. Skannaus löysi HTTP, HTTPS, SSH, portmapper, status ja zeroconf (Rendezvous) -palvelut. Palvelu, josta löytyi eniten haavoittuvuuksia, oli HTTPS. Haavoittuvuuksia löytyi 34. Kuvassa 31 on havainnollistettu tavallisimpia palveluita, joita skannaus löysi palvelimelta. Kuvassa 32 on havainnollistettu palveluja, joista löytyi eniten haavoittuvuuksia.



Kuva 31. Tavallisimmat palvelut.

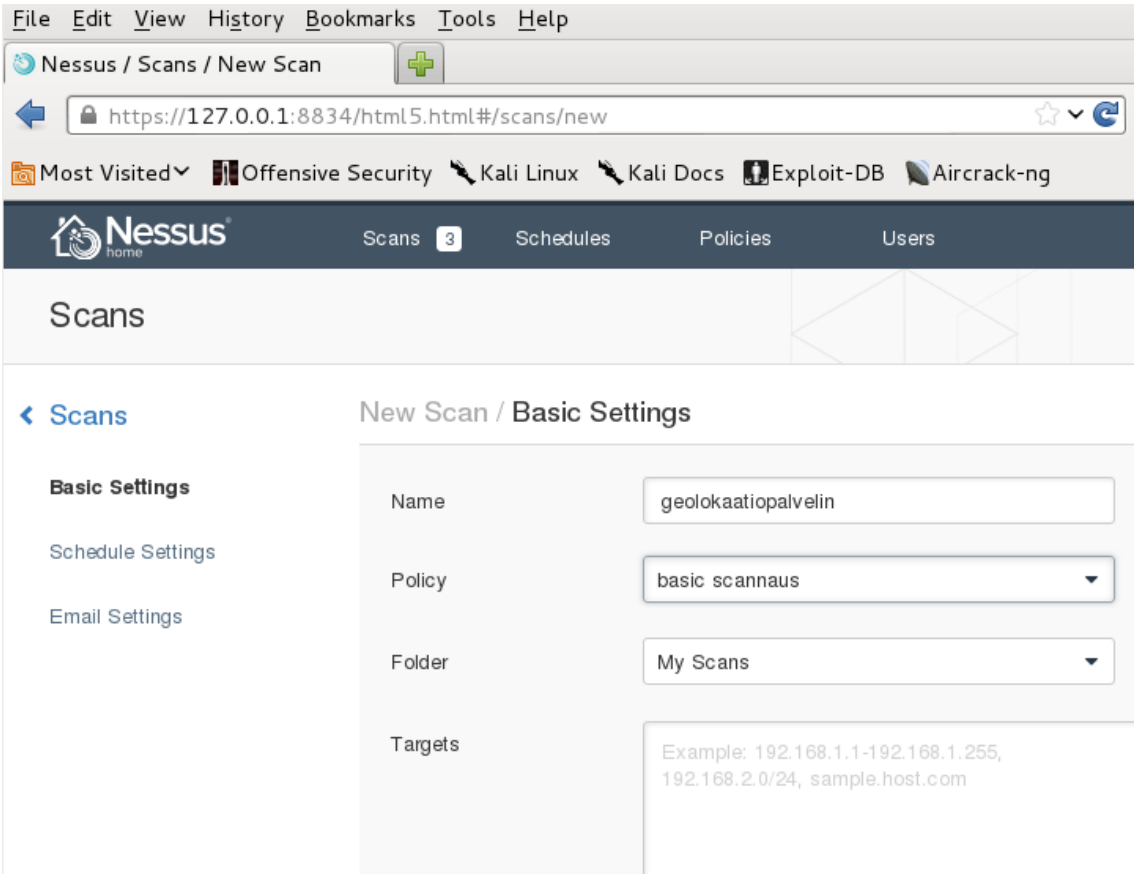
Kuva 32. Palvelut, joista löytyi eniten haavoittuvuuksia.

6.3 Nessus

Nessus haavoittuvuus-skannerilla käyttäjä voi tehdä muun muassa päivitys-, asetus- ja sääntöjen noudattamistarkastuksia. Nessuksen sisältämän skannauksen avulla voidaan etsiä

- haavoittuvuuksia, jotka mahdollistavat hakkerin hallitsevan arkaluonteisia tiedostoja etäisesti
- virhemäärityksiä (esimerkiksi avoin viestinvälityspalvelin, puuttuvat päivitykset jne.)
- salasanoja
- Denial-of-Service -hyökkäykset (Tenable 2014).

Kuvassa 33 on kuvakaappaus Nessuksen käytöstä selaimella.



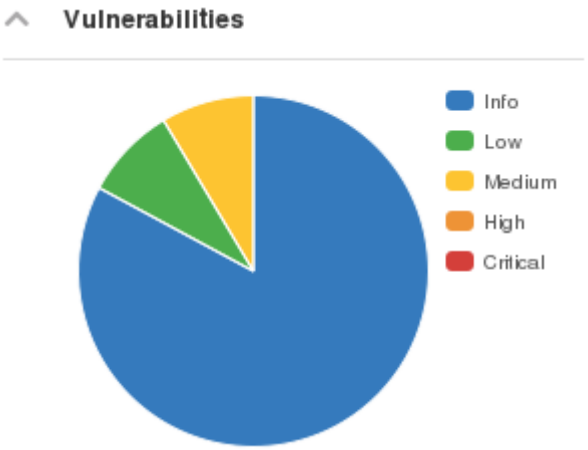
Kuva 33. Nessuksen käyttö selaimella.

Tulokset

Nessuksen haavoittuvuus-skanneri löysi 35 haavoittuvuutta. Mikään näistä ei ollut korkean asteen tai kriittinen haavoittuvuus. Kuvassa 34 on Nessuksen skannauksen löytämiä haavoittuvuuksia geolokaatiopalvelimelta. Kuvassa 35 havainnollistetaan skannauksen löytämien haavoittuvuuksien vakavuustasoa ja määrää geolokaatiopalvelimella.

Severity ▲	Plugin Name	Count
MEDIUM	mDNS Detection (Remote Network)	1
MEDIUM	SSL Certificate Cannot Be Trusted	1
MEDIUM	SSL Self-Signed Certificate	1
LOW	SSH Server CBC Mode Ciphers Enabled	1
LOW	SSH Weak MAC Algorithms Enabled	1
LOW	SSL RC4 Cipher Suites Supported	1
INFO	Nessus SYN scanner	4
INFO	RPC Services Enumeration	4
INFO	Service Detection	4
INFO	Backported Security Patch Detection (WWW)	2
INFO	HTTP Methods Allowed (per directory)	2

Kuva 34. Nessuksen löytämiä haavoittuvuuksia geolokaatiopalvelimelta.



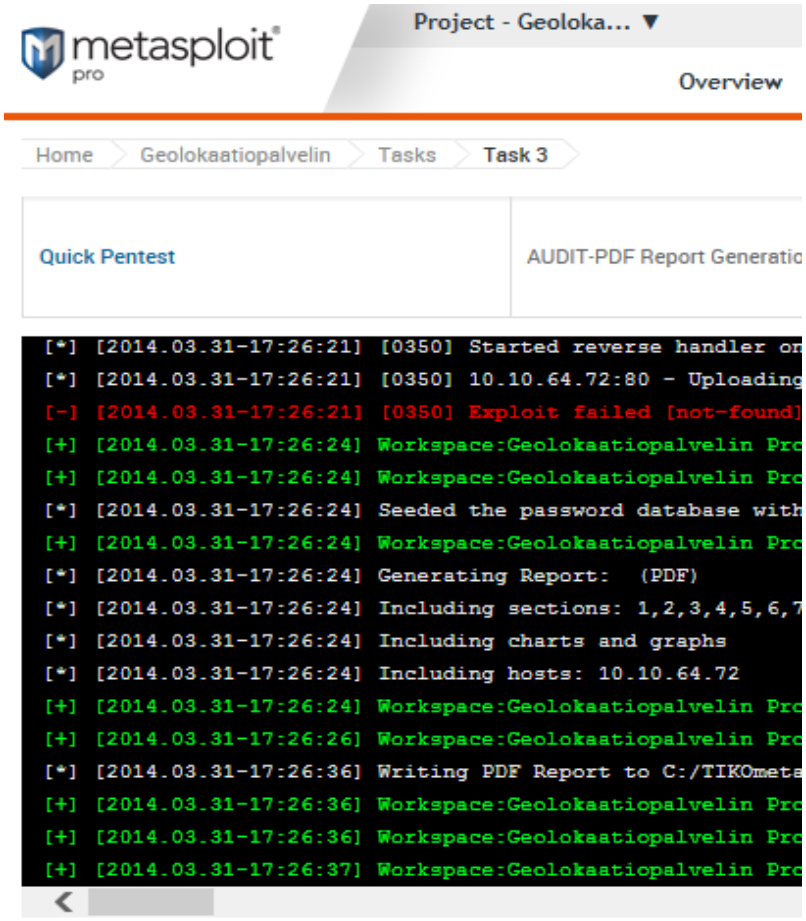
Kuva 35. Nessuksen löytämät haavoittuvuudet, niiden vakavuustaso ja määrä geolokaatiopalvelimella.

6.4 Metasploit Pro

Metasploit Pro skannaa ja hyödyntää web-sovelluksia, suorittaa käyttäjien manipulointikampanjoita ja hakee pääsyä kohdejärjestelmään tai -verkkoon eri keinoin. Se on haavoittuvuus-skanneri, joka

- validoi turvallisuusriskejä osana organisaation haavoittuvuuden hallintaohjelmaa
- simuloi hyökkäyksiä turvallisesti verkossa
- tarkastaa suojaukset ja turvatarkistukset
- mittaa tietoturvaohjelmien tehokkuutta
- tarkastaa salasanaturvallisuuden Windows ja Linux -kirjautumisissa (Rapid7 2014a).

Kuvassa 36 on kuvakaappaus Metasploit Pron käytöstä selaimella.



Kuva 36. Metasploit Pron käyttö selaimella.

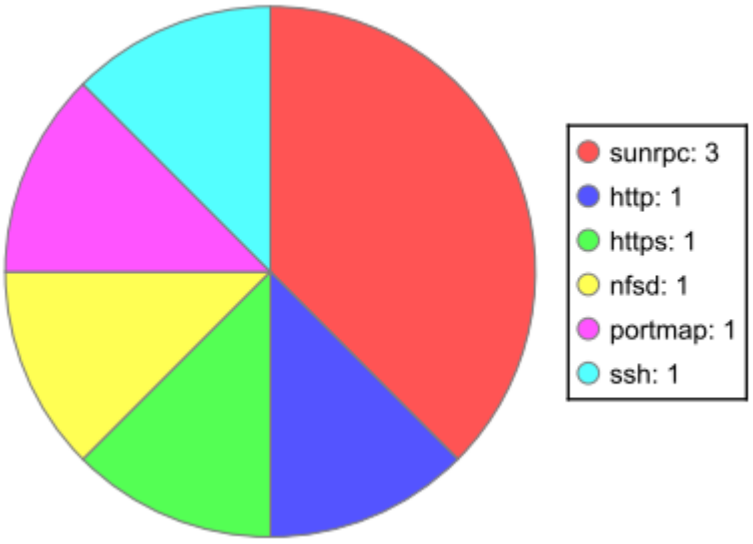
Tulokset

Turvallisuustarkastuksen raportti esittää Metasploit Pro:lla tehtyä skannausta, joka yrittää pintapuolisesti löytää tietoa kohdepalvelimesta. Skannaus löysi kohteesta kahdeksan hyökkäyksille alttiina olevaa palvelua. Yksikään moduuli ei onnistunut hyödyntämään tietoja palvelimelta, eikä yhtään sisäänkirjautumistietoa löydetty. Taulukossa 3 havainnollistetaan skannauksen löytämiä aktiivisia palveluja.

Taulukko 3. Skannauksen löytämät aktiiviset palvelut.

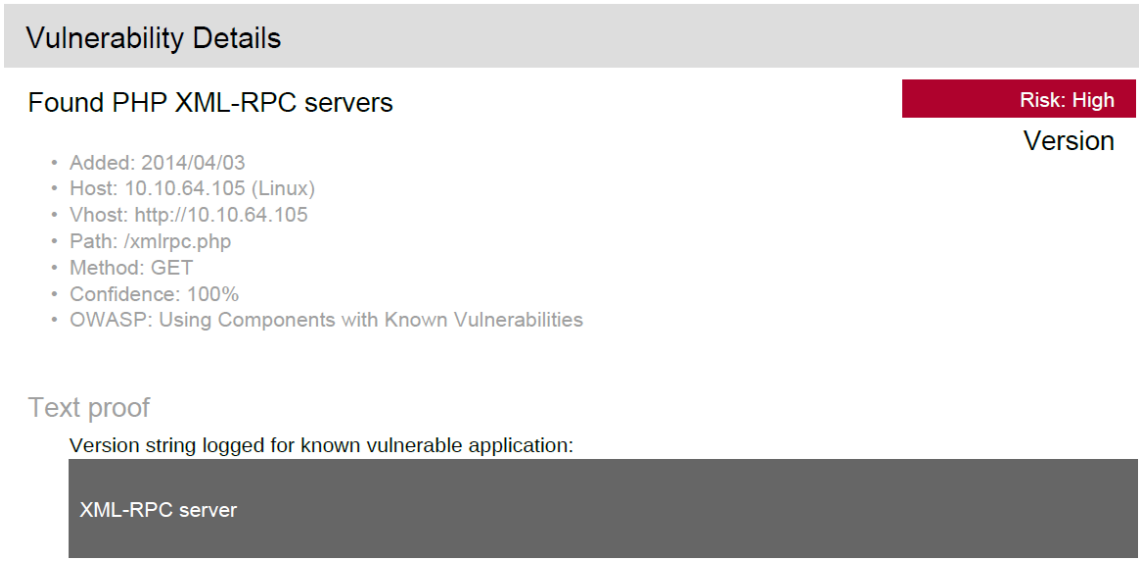
Port	Protocol	Name	Info
22	tcp	ssh	SSH-2.0-OpenSSH_6.0p1 Debian-4
80	tcp	http	Apache/2.2.22 (Debian) (Powered by PHP/5.4.4-14+deb7u5)
111	tcp	sunrpc	100000 v2
111	udp	portmap	100000 v4 TCP(111), 100000 v3 TCP(111), 100000 v2 TCP(111), 100000 v4 UDP(111), 100000 v3 UDP(111), 100000 v2 UDP(111), 100024 v1 UDP(39329), 100024 v1 TCP(42358)
443	tcp	https	Apache/2.2.22 (Debian) (Powered by PHP/5.4.4-14+deb7u5)
2049	udp	nfsd	NFS Daemon 100005 v1
39329	udp	sunrpc	100024 v1
42358	tcp	sunrpc	100024 v1

Skannaus havaitsi kahdeksan käynnissä olevaa palvelua. Kuvassa 37 havainnollistetaan käynnissä olevia palveluja.



Kuva 37. Käynnissä olevat palvelut geolokaatiopalvelimella.

Seuraavaksi suoritetaan Web-sovellusten haavoittuvuutta testaava skannaus. Skannaus löysi yhden haavoittuvuuden, jonka riskitaso on korkea. Skannauksen mukaan sovelluksella nimeltä OWASP on mahdollista hyväksikäyttää haavoittuvuutta. Kuvassa 38 on kuvakaappaus Web-sovellusten haavoittuvuus- skannauksesta.



Kuva 38. Web-sovellusten haavoittuvuusskannaus.

7 HYÖKKÄYKSET METASPLOITILLA

7.1 Yleisesti käytettyjä hyökkäysyrityksiä

Seuraavassa vaiheessa tutkitaan, kuinka palvelin reagoi muutamiin yleisimpiin Metasploit-hyökkäyksiin. Oletuksena on, että hyökkäykset eivät onnistu murtautumaan kohdekoneelle, tai kaappaamaan minkäänlaista tietoa. Samalla testataan palvelimen toteutusta hyökkäämällä myös palveluihin, joiden ei pitäisi olla päällä. Tarkoituksena on myös opetella Metasploitin käyttöä yleisesti. Alla on listattuna Metasploitissa suoritettavat hyökkäykset, joita testauksessa aiotaan käyttää:

- postgres_login
- unreal_ircd_3281_backdoor
- vsftpd_234_backdoor
- f5_bigip_known_privkey
- symantec_smg_ssh
- tectia_passwd_changereq
- drb_remote_codeexec
- usermap_script
- distcc_exec
- java_rmi_server.

PostgreSQL on avoimena lähdekoodina jaettava olio-relaatiotietokantapalvelin (tietokannan hallintajärjestelmä), joka on lisensoitu joustavalla BSD-tyyppisellä lisenssillä (PostgreSQL 2014). Vaikka postgres-palvelua ei löydy avoimista porteista, suoritetaan silti skannaus sen löytämiseksi. Käytetään Metasploit-työkalua postgres-palvelun etsimisessä.

Aluksi aukaistaan Metasploit manuaalisesti ja syötetään seuraavat komennot:

msfconsole

search postgresql

Haku listaa apumuuttujia ja hyödynnettäviä palveluja. Haku löysi apumuuttujan nimeltä *”auxiliary/scanner/postgres/postgres_login”.* Lisäksi siinä lukee *”PostgreSQL Login Utility”*.

Yritetään hyväksikäyttää apumuuttujaa kirjautumalla siihen brute force -menetelmällä käyttämällä oletuskäyttäjätunnuksia ja salasanoja. Käytetään komentoa:

msf > use auxiliary/scanner/postgres/postgres_login

Asetetaan kohdekone:

msf auxiliary(postgres_login) > set RHOSTS 10.10.64.72

Suoritetaan komento:

msf auxiliary(postgres_login) > run

Vaikka yhteyskokeilu kohdekoneeseen onnistuukin, yritykset luoda yhteyksiä aikakatkaistaan.

Seuraavaksi testattiin laittaa VirtualBoxista Kali-Linuxin verkkoasetuksista sillattu yhteys kohdekoneen kanssa. Tavoitteena on, että yhteyskokeilu Kali-Linux-koneen ja geolokaatiopalvelimen välillä onnistuu.

Käynnistetään Kali-Linux uudelleen. Käynnistämisen jälkeen koetetaan aikaisempia komentoja uudelleen.

Haku ei kykene löytämään postgres-tunnuksia. Yritykset luoda yhteyksiä kohdekoneeseen eivät onnistu, koska kohdekone estää jokaisen yrityksen.

Moduuli unreal_ircd_3281_backdoor hyödyntää takaovea, joka syötettiin Unreal IRCD 3.2.8.1-version latausarkistoon (Rapid7 2014b). Yritetään hyökätä kohdekoneeseen unreal_ircd_3281_backdoor-moduulilla. Käytetään komentoja:

use exploit/unix/irc/unreal_ircd_3281_backdoor

set RHOST 10.10.64.72

exploit

Kohdekone esti hyökkäysyrityksen.

Moduuli VSFTPD v2.3.4 hyödyntää takaovea, joka syötettiin VSFTPD:n latausarkistoon (Rapid7 2014c). Yritetään hyökätä kohdekoneeseen vsftpd_234_backdoor-moduulilla. Käytetään komentoja:

use exploit/unix/ftp/vsftpd_234_backdoor

set RHOST 10.10.64.72

exploit

Kohdekone esti hyökkäysyrityksen.

F5 Networks on maailman johtava ADC (Application Delivery Controller) -laitetoimittaja. F5 BIG-IP ADC -laitteisto toimii välityspalvelimena ADC:llä julkaistavien palvelujen edessä (DCC 2014). F5-laite voi vahingossa toimittaa pyytäjälle staattisen ssh-avaimen, jota voidaan käyttää root-käyttäjän tunnistautumiseen monessa BigIP-laitteissa. F5-laitteet syöttävät julkisen tai salaisen avainparin BIG-IP-laitteisiin, ja joiden avulla kirjautuminen ilman salasanaa on mahdollista mihin tahansa BIG-IP-laitteeseen. Koska avain on helposti haettavissa, hyökkääjä voi käyttää sitä luvatta etäiseen kirjautumiseen root-käyttäjänä (Rapid7 2014e). Yritetään hyökätä kohdekoneeseen f5_bigip_known_privkey-moduulilla. Tunnistautuminen epäonnistui.

Moduuli symantec_smg_ssh hyödyntää huonosti asetettua oletusasetusta Symantecin viestintäportissa. ”Support”-käyttäjällä on tunnettu oletussalasana, jota voidaan käyttää SSH-palvelun sisäänkirjautumiseen. Tämän lisäksi on mahdollista saada käyttöön suoritusoikeuksia. (Rapid7 2014f). Yritetään hyökätä kohdekoneeseen symantec_smg_ssh-moduulilla. Tunnistautuminen epäonnistui jälleen.

Moduuli tectia_passwd_changereq hyödyntää haavoittuvuutta, jonka voi löytää Tectian SSH-palvelimessa Unix-pohjaisissa käyttöjärjestelmissä. Ohjelmointivirhe aiheutuu SSH2_MSG_USERAUTH_PASSWD_CHANGEREQ-pyynnöstä ennen salasanatodennusta, ja se sallii minkä tahansa käyttäjän

ohittaa sisäänkirjautumisen ja saamaan pääkäyttäjän oikeudet (Rapid7 2014g). Yritetään hyökätä kohdekoneeseen tectia_passwd_changereq-moduulilla. Tunnistautuminen epäonnistui jälleen.

Moduuli drb_remote_codeexec hyödyntää koodin suorittamisen haavoittuvuuksia dRuby:ssä (Rapid7 2014h). Yritetään hyökätä kohdekoneeseen drb_remote_codeexec-moduulilla. Kohdekone esti yrityksen jälleen.

Moduuli usermap_script hyödyntää komentojen suorittamisen haavoittuvuutta Samban versiosta 3.0.20 aina versioon 3.0.25rc3 asti, kun käytetään asetusta ”username map script”. Hyökkääjä kykenee suorittamaan toimintoja, kun käyttäjänimi määritellään shellin metamerkeillä. Metamerkeillä tarkoitetaan seuraavia merkkejä: *; & ( )* | < > \ *$ * rivinvaihto välilyönti tabulaattori*. Tämän haavoittuvuuden hyödyntämisessä ei tarvitse tunnistautumista, koska sitä käytetään käyttäjänimien kartoittamisessa ennen käyttäjän tunnistautumista (Rapid7 2014i). Yritetään hyökätä kohdekoneeseen käyttämällä moduulia usermap_script. Käytetään komentoja:

use exploit/multi/samba/usermap_script

set RHOST 10.10.64.72

exploit

Kohdekone esti kuitenkin yhteyden muodostamisen.

Moduuli distcc_exec käyttää dokumentoitua haavoittuvuutta suorittaakseen komentoja missä tahansa järjestelmässä, joka suorittaa distccd-työkalua (Rapid7 2014j). Yritetään hyökätä kohdekoneeseen moduulilla usermap_script. Käytetään komentoja:

use exploit/unix/misc/distcc_exec

set RHOST 10.10.64.72

exploit

Kohdekone esti yhteysyrityksen.

RMI (Remote Method Invocation) on sovellusohjelmoinnin rajapinta, jota käytetään Java-ympäristöissä. Moduuli java_rmi_server hyödyntää RMI-rekisterin ja RMI-aktivoinnin palvelujen oletusasetuksia, jotka sallivat luokan lataamisen miltä tahansa etäiseltä URLilta. Sitä voidaan käyttää esimerkiksi RMI-rekisteriä vastaan. Näiden lisäksi sitä voidaan käyttää myös useampia RMI-laitteistoja vastaan. RMI-metodikutsujat eivät tue tai vaadi minkäänlaista tunnistautumista (Rapid7 2014k). Yritetään hyökätä kohdekoneeseen moduulilla java_rmi_server. Käytetään komentoja:

use exploit/multi/misc/java_rmi_server

set RHOST 10.10.64.72

exploit

Kohdekone esti yhteysyrityksen.

Kuten oletettiin, käytetyt hyökkäysmetodit eivät onnistuneet murtautumaan kohdekoneeseen, tai kaappaamaan siitä minkäänlaista tietoa. Ylimääräisiä päällä olevia palveluja ei testauksen perusteella löytynyt.

7.2 PHP-haavoittuvuus

Haavoittuvuusskannauksessa Nexpose löysi kohteesta kolme kriittistä PHP-haavoittuvuutta. Yritetään hyödyntää haavoittuvuutta hyökkäämällä siihen moduulilla php_cgi_arg_injection. Kun PHP:ta suoritetaan CGI:n versioilla 5.3.12 tai 5.4.2, se on haavoittuvainen muuttujainjektiohaavoittuvuudelle. Moduuli php_cgi_arg_injection hyödyntää php.ini-tiedoston -d-lippua koodin suorittamiseen. (Rapid7 2014d.)

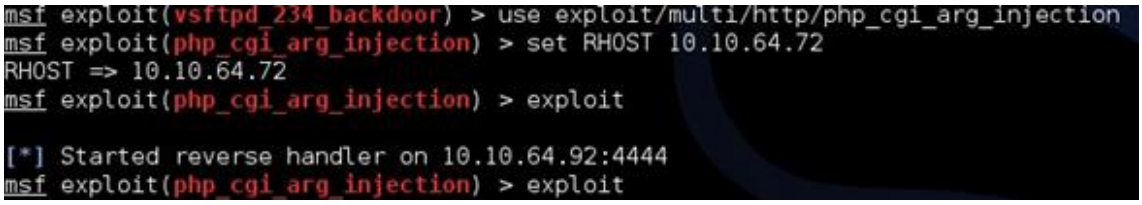
Yritetään hyökätä kohdekoneeseen php_cgi_arg_injection-moduulilla. Käytetään komentoja:

use exploit/multi/http/php_cgi_arg_injection

set RHOST 10.10.64.72

exploit

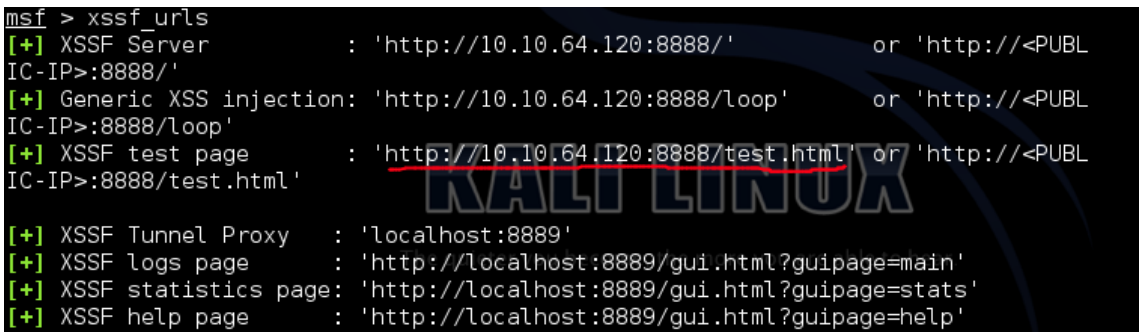
Hyökkäys ei onnistunut. Kuvassa 39 on kuvakaappaus yrityksestä.



Kuva 39. Hyökkäys moduulilla php_cgi_arg_injection.

7.3 XSS-injektio

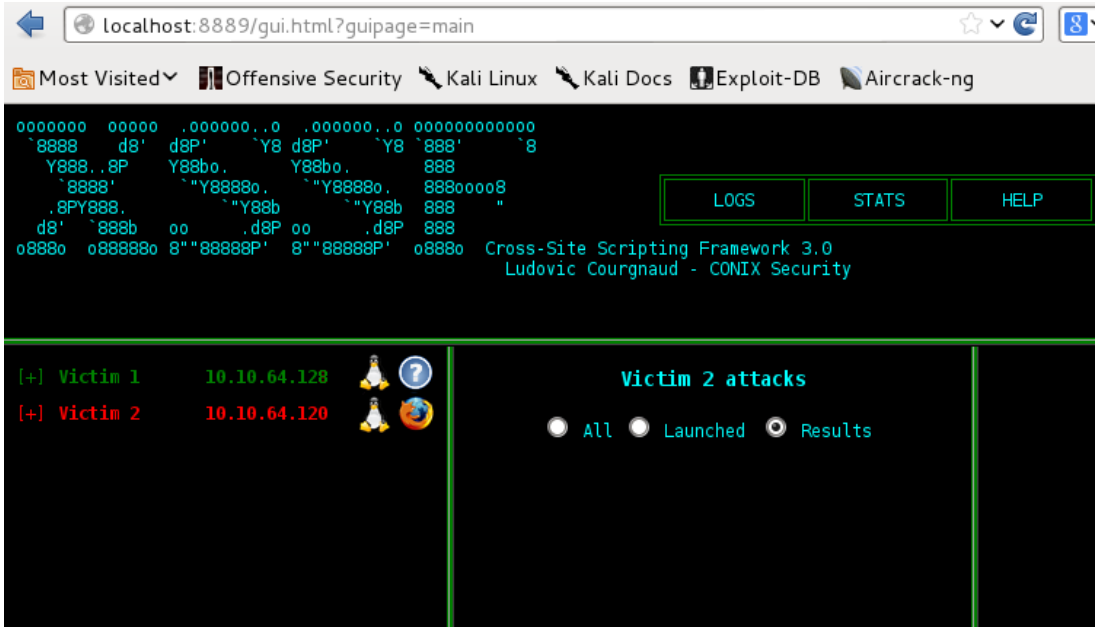
Tarkoituksena on hyödyntää OWASP ZAPin haavoittuvuusskannerin löytämää vakavaksi luokiteltua XSS-haavoittuvuutta. Aluksi asennetaan XSSF-työkalu metasploitiin. Luodaan XSS-injektio kohdekonetta varten. Jos kohdekone avaa XSS-injektiota varten luodun osoitteen selaimellaan, on mahdollista saada siitä tietoa, sekä hyökätä siihen eri keinoin. Kuvassa 40 havainnollistetaan XSSF-työkalulla luotuja osoitteita XSS-injektiota varten.



Kuva 40. Osoitteiden luominen XSS-injektiota varten.

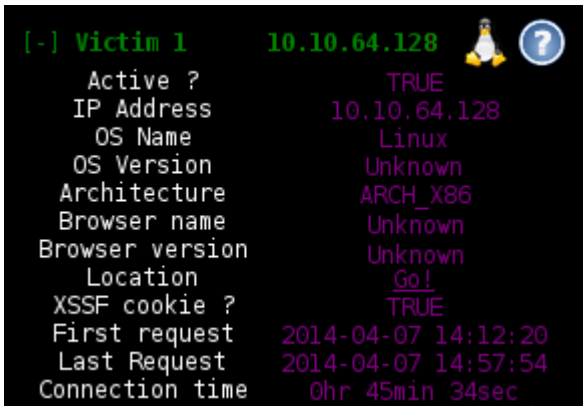
Oletetaan, että kohdekone avaa linkin ”http://10.10.64.120:8888/test.html, joka on syötetty sille esimerkiksi sosiaalisen median kautta.

Seuraavaksi avataan XSSF-ohjelman luoma sivusto, jossa voi tarkastella kohdekoneiden tietoja. Kuvassa 41 on havainnollistettu sivuston ulkoasua.



Kuva 41. XSSF-ohjelman tilastosivusto.

Kuvassa 42 on kuvakaappaus kohdekoneen tietojen tarkastelusta. Näiden tietojen avulla hyökkääjä voi esimerkiksi valita, minkälaisia hyökkäyksiä aiotaan toteuttaa. Hyökkäyksien valitseminen voi riippua esimerkiksi kohdekoneen käyttöjärjestelmästä, sen versiosta ja arkkitehtuurista.



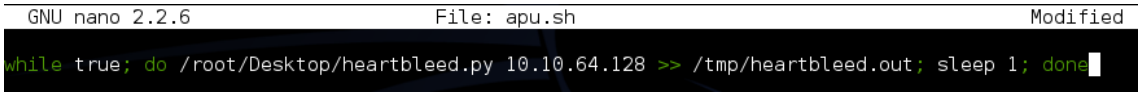
Kuva 42. Geolokaatiopalvelimesta löydettyjä tietoja.

7.4 OpenSSL Heartbleed

Seuraavaksi yritetään käyttää hyödyksi ohjelmointivirhettä, jota kutsutaan nimellä ”OpenSSL Heartbleed”. Sen löysi suomalainen IT-yritys nimeltä Codenomicon ja Googlen tietoturvaosastolla työskentelevä tutkija Neel Mehta. Heartbleed on vakava haavoittuvuus OpenSSL:n ohjelmiston kirjastossa. Tämä haavoittuvuus mahdollistaa tietojen varastamisen SSL/TLS:n suojaamista kohteista. SSL/TLS tarjoaa viestinnän turvallisuutta ja yksityisyyttä sovelluksille, kuten esimerkiksi web-, sähköposti-, pikaviesti ja joillekin virtuaalisille yksityisille verkoille (kuten esimerkiksi VPN).

Heartbleed-ohjelmointivirhe sallii kenen tahansa käyttäjän lukea järjestelmien muisteja, jotka käyttävät OpenSSL:n haavoittuneita versioita. Tämä mahdollistaa salatuttujen avaimien löytämisen suoraan palveluista. Sen kautta on mahdollista saada selville muun muassa verkkosivujen käyttäjien käyttäjänimet ja salasanat. Niitä voidaan käyttää esimerkiksi tietomurtoihin tai käyttäjänä esiintymiseen (Heartbleed 2014). Heartbleedillä pystytään kaappaamaan tietoja noin 64 kilobitin verran per suoritus.

Testiä varten verkosta ladataan Python-kielellä suoritettava ohjelmakoodi heartbleed.py, ja luodaan bash-scriptillä toimiva komento erilliseen tiedostoon. Komento ”while true” tekee silmukan, joka toistaa syötetyt komennot, kunnes ehto täyttyy. Komento ”do heartbleed.py” suorittaa python-koodia osoitteeseen 10.10.64.128. Komento ”>> /tmp/heartbleed.out;” kirjoittaa tiedostoon ”heartbleed.out” löydettyjä tietoja. Kuvassa 43 on kuvakaappaus komennon luomisesta erilliselle tiedostolle.



Kuva 43. Luotu bash-script-komento tiedoston sisällä.

Annetaan luodulle tiedostolle suoritusoikeudet komennolla *chmod +x apu.sh*.

Suoritetaan apu.sh komennolla *./apu.sh*.

Tutkitaan löydettyjä tietoja, joita ohjelmakoodi jatkuvasti kerää tiedostoon heartbleed.out. Ohjelmakoodi ilmoittaa palvelimen 10.10.64.128 olevan haavoittuvainen. Kuvassa 44 on pieni osa ohjelmakoodin löytämistä tiedoista.

```
root@kalilinux:/tmp# tail -f heartbleed.out
WARNING: 10.10.64.128:443 returned more data than it should - server is vulnerab
le!
#################################################################

.@....SC[...r....+..H...9...
....w.3....f...
...!.9.8........5..............
.........3.2.....E.D...../...A...............................I.........
..........
..................................#.........................................
....


#################################################################
Connecting to: 10.10.64.128:443 with TLSv1.1

Sending Client Hello...
Sending heartbeat request...
WARNING: 10.10.64.128:443 returned more data than it should - server is vulnerab
le!
#################################################################

.@....SC[...r....+..H...9...
```

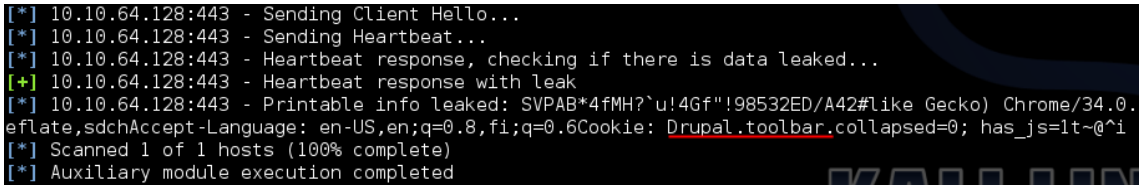
Kuva 44. Palvelin 10.10.64.128 on haavoittuvainen ohjelmakoodille.

Heartbleed löysi kohdekoneelta hakemistoja, joista löytyy muun muassa moduuleja. Kuvassa 45 on kuvakaappaus löydetyistä hakemistoista.

```
*]Q9}`4cL('om]pWh^9~!!w!Ygx?AQlQN]`6k~xq^Z+<F''Jw1u^K4a!5&I8|-1+F%IFp&IF`FF`$F`$F`$F`$F1pF>~
Fq=9uH,F(H,F(WS/home/wise/pulic_html/modules/contextualZ,F:,F:WS/home/wise/pulic_html/module
s/contextual/contextual.moduleQN~FF1^FFMNFPpqiE-F'-F'WS/home/wise/pulic_html/modules/dashboa
rd.FY|-F8-F8WS/home/wise/pulic_html/modules/dashboard/dashboard.module |FcF@.F@.F@"FpFF/F#F,
F,Fx"F%F,FhV.FFFF`"F3.F"
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(openssl_heartbleed) > run
```

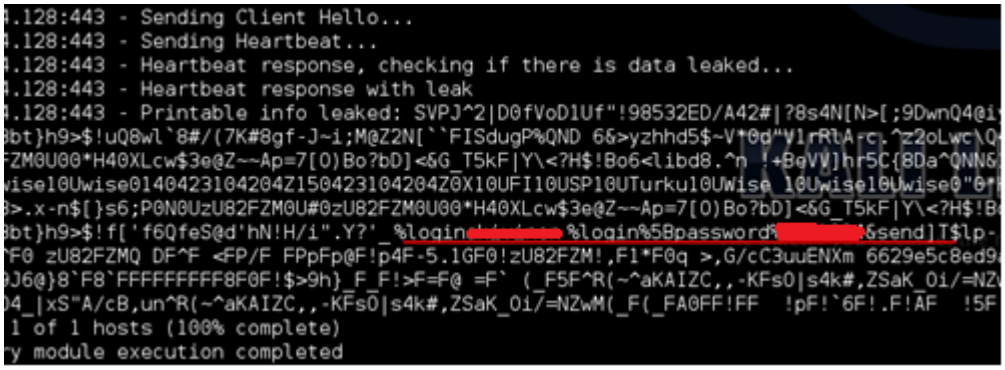
Kuva 45. Heartbleedin löytämiä hakemistoja.

Palvelin näyttää käyttävän Drupal-sisällönhallintajärjestelmää. Kuvassa 46 on kuvakaappaus havainnosta.



Kuva 46. Drupal-työkalupakin löytäminen.

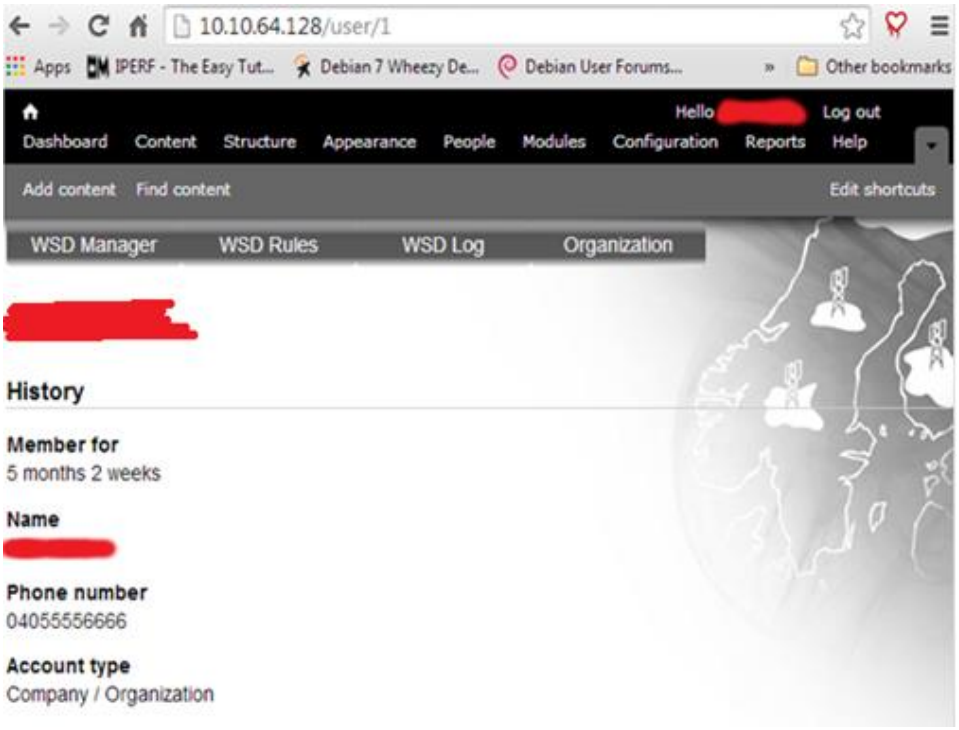
Heartbleed löysi käyttäjätunnuksen ja salasanan. Kuvassa 47 on kuvakaappaus hyökkäyksestä.



Kuva 47. Mahdollisen käyttäjätunnuksen ja salasanan löytäminen.

Tämän jälkeen kokeiltiin löydettyä käyttäjätunnusta ja salasanaa suoraan palvelimen kirjautumisjärjestelmään.

Kirjautuminen onnistui. Käyttäjätunnuksella pystyy muokkaamaan kaikkia Drupalin asetuksia. Löydetty käyttäjätunnus toimii todennäköisesti ylläpitäjän oikeuksilla. Kuvassa 48 on kuvakaappaus sisäänkirjautumisesta.



Kuva 48. Onnistunut sisäänkirjautuminen.

8 TULOSTEN TARKASTELU

Opinnäytteessä käytettyjen työkalujen perusteella palvelimen turvataso oli liian matala suojautumaan seuraavilta hyökkäyksiltä:

- Cross-Site scripting (XSS)
- SSL -protokollaan kohdistuvat hyökkäykset.

Haavoittuvuusskannerien antamien tulosten perusteella palvelimen turvataso saattaa olla liian alhainen suojautumaan seuraavilta hyökkäyksiltä:

- SSH -protokollaan kohdistuvat hyökkäykset
- Useat web-sovelluksiin kohdistuvat hyökkäykset
- HTML:ään kohdistuvat hyökkäykset
- Apache-palvelinohjelmaan kohdistuvat hyökkäykset
- SQL-hyökkäykset
- PHP-pohjaisten sovelluksiin kohdistuvat hyökkäykset (kuten esimerkiksi XML-RPC).

Vaikka opinnäytteessä suoritettuja testauksia ei voi pitää täysin kattavina testauksina, palvelimen tietoturvasta löytyy tietoturva-aukkoja usealta eri protokollakerrokselta. Jotta tietoturvan tarkastelu olisi ollut laajempi, olisi jo löytyneiden hyökkäysvektoreiden lisäksi tutkittava muita eri reittejä murtautua palvelimelle. Jos tietoturvan testaaminen olisi jätetty vain haavoittuvuusskannereiden varaan, olisi muun muassa ylläpitäjän käyttäjätunnus ja salasana jäänyt löytämättä. Tämän vuoksi tunkeutumistestaus on hyvä keino palvelimen tai verkon tietoturvan kokonaisvaltaista testaamista varten.

Hyökkäyksiä ennaltaehkäiseviä keinoja on muun muassa pitää kaikkien sovellusten päivitykset ajan tasalla. Käyttäjätunnusten ja salasanojen tulisi olla riittävän monimutkaisia, jottei mahdollinen hyökkääjä voisi esimerkiksi brute force -murtotekniikalla löytää oikeita tunnuksia.

PHP:sta ilmoitetut haavoittuvuudet voidaan korjata päivittämällä PHP uusimpaan versioon. XSS-hyökkäyksiä vastaan on hankala suojautua, mutta on olemassa ainakin yksi hyvä keino niiden ennaltaehkäisemiseksi: HTML-dokumenttiin ei kannata syöttää ylimääräistä tai arkaluonteista tietoa. SQL-injektioita vastaan kannattaisi muun muassa laskea niiden käyttäjätunnusten oikeuksia, joilla on pääsy tietokantaan. Heartbleed-injektiota varten kannattaa päivittää OpenSSL uusimpaan versioon, sillä versiot 1.01 - 1.0.1f ovat haavoittuvaisia. Uusin versio ei ole haavoittuvainen Heartbleed-injektiolle.

9 POHDINTA

Opinnäytetyön tekeminen alkoi siitä, kun minut kutsuttiin mukaan WISE-projektiin. WISE-projektin yksi ongelmista oli, että siinä käytettävän geolokaatiopalvelimen tietoturvaa ei oltu syvällisemmin testattu. Otin vastuulleni tunkeutumistestata kyseisen geolokaatiopalvelimen, mistä lopulta muodostuikin opinnäytetyöni aihe.

Opinnäytetyön alkuperäinen tavoite oli suorittaa tunkeutumistestaus geolokaatiopalvelimelle, joka on käytössä WISE-projektissa. Koska lupaa levykuvan suoralle kopioimiselle ei saatu sen tietokannan sisältävän luottamuksellisen tiedon vuoksi, oli luotava itse mahdollisimman tarkasti vastaavanlainen palvelin suojattuun ympäristöön. Opinnäytetyön pääasiallinen tarkoitus oli mukailla mahdollisen ulkopuolisen hyökkääjän käyttämiä hyökkäysmenetelmiä ja onnistuneesti murtautua palvelimelle. Lisäksi tavoitteena oli muokata käyttäjäoikeuksia sekä muokata ja tallentaa geolokaatiopalvelimen tiedostoja.

Tunkeutumistestaukset ovat hyvä tapa tunnistaa haavoittuvuuksia järjestelmistä ja verkoista, joihin on jo valmiiksi asennettu turvatoimia. Tunkeutumistestauksessa on tarkoituksena simuloida keinoja, joita mahdollinen hyökkääjä voisi käyttää pyrkiessään murtautumaan tietoverkkoihin ja -järjestelmiin. Tunkeutumistestauksen laajuuteen, luotettavuuteen ja laatuun vaikuttaa käytettävissä oleva aika, resurssit ja testausta suorittavien henkilöiden taito. Testien tulokset dokumentoidaan ja lopputulos esitetään järjestelmän omistajalle raporttina. Raportin pohjalta tunnistettuja haavoittuvuuksia voidaan alkaa paikkaamaan.

Tunkeutumistestaamista varten luotuja ilmaisia ja kaupallisia työkaluja on tarjolla paljon. Työkaluja käyttämällä testauksen eri vaiheita kyetään automatisoimaan, joka nopeuttaa prosessia. Tässä työssä tehdyt testaukset suojatussa laboratorioverkossa ilmentävät kuinka helppokäyttöisiä ja toimivia tämän päivän työkalut ovat. Suojatussa sisäverkossa suoritetut testaukset

poistavat kuitenkin ainakin yhden oleellisen, tietoturvaan vaikuttavan tekijän, ihmisen. Tietoverkon tai -järjestelmän tietoturvaa tutkittaessa on muistettava, että suuri osa tietoturvariskeistä johtuu inhimillisistä seikoista. Näin ollen työkaluja, jotka luottavat ihmisen tekemiin virheisiin, ihmisen manipulointiin tai hakukoneiden antamiin tietoihin, ei voida käyttää.

Opinnäytetyön teoriaosuudessa tutkittiin mikä on hakkeri, miten heidät yleisesti luokitellaan, mitä menetelmätapoja hakkerit käyttävät ja kuinka käyttäjä voi suojautua hakkerointia vastaan. Tämän lisäksi opinnäytetyössä määriteltiin mikä on tunkeutumistestaus, mitkä ovat sen eri tyypit ja testaamisen eri vaiheet. Teoriaosuuden lopussa tutkittiin, mitä erilaisia sovelluksia ja työkaluja voidaan tunkeutumisestaamisen tiedonkeruuvaiheessa käyttää.

Geolokaatiopalvelimesta luotiin mahdollisimman yksityiskohtaisesti oikeaa vastaava palvelin, joka liitettiin Turun Ammattikorkeakoulun suojattuun laboratorioverkkoon. Tämän jälkeen palvelin tutkittiin erilaisilla työkaluilla ja haavoittuvuusskannereilla mahdollisten haavoittuvuuksien löytämiseksi.

Mielestäni opinnäytetyö on onnistunut vaatimuksiltaan, joita sille oli asetettu. Geolokaatiopalvelimesta löydettiin haavoittuvuuksia. Onnistuin hyväksikäyttämään löytämiäni haavoittuvuuksia. Löysin palvelimelta arvokasta tietoa hyökkäyksien avulla. Onnistuin kirjautumaan Drupal-sisällönhallintajärjestelmään. Onnistuneesta heartbleed-hyökkäyksestä ei jäänyt lokitietoihin jälkiä. Opinnäytteessä kerrotaan, kuinka löydetyiltä haavoittuvuuksilta voidaan suojautua ja kuinka niitä voidaan paikata. Tulevaisuuden kannalta tätä työtä kannattaisi jatkaa muun muassa etsimällä uusia keinoja murtautua palvelimelle, muuttaa palvelimen käyttäjätunnuksien oikeuksia, muokata ja tallentaa tiedostoja sekä yrittää selvittää, kuinka paljon palvelimen lokitietoihin jää jälkiä hyökkäysyrityksistä.

Olen tyytyväinen omaan työpanostukseeni projektissa. Olen oppinut projektia tehdessäni paljon uutta, kuten esimerkiksi tunkeutumistestaamista, hakkeroinnin perusteita, tiedon hakua oikeista paikoista, virtualisointia, organisointia, dokumentointia ja kärsivällisyyttä. Huomasin, että

tunkeutumistestaaminen voi olla hyvinkin aikaa vievää. Pyrin aina tutkimaan asioita perusteellisesti ennen kuin toimin. Ratkaisujen löytäminen itse oli palkitsevaa vaikkakin aikaa vievää.

LÄHTEET

About 2014. What Is A Rootkit? Viitattu 8.4.2014. http://netsecurity.about.com/od/frequentlyaskedquestions/f/faq_rootkit.htm.

Acunetix 2013. Web Vulnerability Scanner v9. Viitattu 31.3.2014. http://www.acunetix.com/vulnerability-scanner/wvsmanual.pdf.

Acunetix 2014a. Cross Site Scripting Attack. Viitattu 8.4.2014. https://www.acunetix.com/websitesecurity/cross-site-scripting/.

Acunetix 2014b. Audit Your Website Security with Acunetix Web Vulnerability Scanner. Viitattu 30.3.2014. http://www.acunetix.com/vulnerability-scanner/.

Allen, L. 2012. Advanced Penetration Testing for Highly-Secured Environments: The Ultimate Security Guide. Birmingham: Packt Publishing.

Answers 2014. What is wormhole attack? Viitattu 8.4.2014. http://wiki.answers.com/Q/What_is_wormhole_attack?#slide=3.

Antivirus 2013. Computer Hackers. Viitattu 7.2.2014 http://www.antivirus.com/security-software/definition/computer-hackers/index.html.

Bhardwaj M. & Singh G.P. 2014. Types of Hacking Attack and their Counter Measure. Viitattu 9.2.2014 http://www.ripublication.com/ijepa/ijepav1n1_7.pdf.

Catb 2014. Hacker. Viitattu 7.2.2014. http://www.catb.org/jargon/html/H/hacker.html.

Crootof, R.; Hathaway, O.; Levitz, P.; Nix, H.; Nowlan, A.; Perdue, W. & Spiegel J. 2011. The Law of Cyber-Attack. Viitattu 9.2.2014 http://www.law.yale.edu/documents/pdf/cglc/LawOfCyberAttack.pdf.

DCC 2014. F5 -tuotteet. Viitattu 14.5.2014. http://www.dcc.fi/index.php/ct-menu-item-3/f5-networks.

Drupal 2014. About Drupal. Viitattu 8.4.2014. https://drupal.org/about.

Edge-Security 2014. The metadata collector. Viitattu 31.3.2014. http://www.edge-security.com/metagoofil.php.

Engebretson, P. 2011. The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy. Waltham: Syngress.

Fast and easy hacking 2014. About Armitage. Viitattu 23.11.2013. http://www.fastandeasyhacking.com/manual#0.

Heartbleed 2014. Heartbleed Bug. Viitattu 11.4.2014. https://heartbleed.com/.

ISS 2014. Spoofing. Viitattu 29.4.2014. http://www.iss.net/security_center/advice/Underground/Hacking/Methods/Technical/Spoofing/default.htm.

Iwriteiam 2014. Definition of a hacker. Viitattu 7.2.2014. http://www.iwriteiam.nl/HackerDef.html.

Klevinsky, T.J., Laliberte, S. & Gupta, A. 2002. Hack I.T. - Security Through Peneration Testing. Boston Pearson Education, Inc.

Linuxmanpages 2014. Options. Viitattu 3.4.2014. http://www.linuxmanpages.com/man1/nmap.1.php.

McAfee 2014. SiteDigger v3.0 Released 12/01/2009. Viitattu 31.3.2014. http://www.mcafee.com/us/downloads/free-tools/sitedigger.aspx.

Melmeg, A. 2014. Penetration Testing. Viitattu 20.2.2014. http://www.giac.org/cissp-papers/197.pdf.

OWASP 2014. Owasp Zed Attack Proxy Project. Viitattu 7.4.2014. https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project.

Paterva 2014. Maltego. Viitattu 31.3.2014. https://www.paterva.com/web6/products/maltego.php.

PC & Tech Authority 2014. FOCA Free 3.0. Viitattu 31.3.2014. http://downloads.pcauthority.com.au/article/22211-foca_free.

PCTools 2014. What are crackers and hackers?. Viitattu 7.2.2014. http://www.pctools.com/security-news/crackers-and-hackers/.

Penetration test 2014. Wikipedia. Viitattu 13.2.2014. http://en.wikipedia.org/wiki/Penetration_test.

Pen-Tests 2014. Penetration Testing vs Ethical Hacking. Viitattu 13.2.2014. http://www.pen-tests.com/penetration-testing-vs-ethical-hacking.html.

Phishing 2014. How to recognize phishing email messages, links, or phone calls. Viitattu 8.4.2014. http://www.microsoft.com/security/online-privacy/phishing-symptoms.aspx.

PostgreSQL 2014. About. Viitattu 1.4.2014. http://www.postgresql.org/about/.

Rapid7 2014a. Use our penetration testing software to. Viitattu 31.3.2014. http://www.rapid7.com/products/metasploit/editions-and-features.jsp.

Rapid7 2014b. UnrealIRCD 3.2.8.1 Backdoor Command Execution. Viitattu 1.4.2014. https://www.rapid7.com/db/modules/exploit/unix/irc/unreal_ircd_3281_backdoor.

Rapid7 2014c. VSFTPD v2.3.4 Backdoor Command Execution. Viitattu 1.4.2014. http://www.rapid7.com/db/modules/exploit/unix/ftp/vsftpd_234_backdoor.

Rapid7 2014d. PHP CGI Argument Injection. Viitattu 1.4.2014. http://www.rapid7.com/db/modules/exploit/multi/http/php_cgi_arg_injection.

Rapid7 2014e. F5 BIG-IP SSH Private Key Exposure. Viitattu 1.4.2014. http://www.rapid7.com/db/modules/exploit/linux/ssh/f5_bigip_known_privkey.

Rapid7 2014f. Symantec Messaging Gateway 9.5 Default SSH Password Vulnerability. Viitattu 1.4.2014. https://www.rapid7.com/db/modules/exploit/linux/ssh/symantec_smg_ssh.

Rapid7 2014g. Tectia SSH USERAUTH Change Request Password Reset Vulnerability. Viitattu 1.4.2014. http://www.rapid7.com/db/modules/exploit/unix/ssh/tectia_passwd_changereq.

Rapid7 2014h. Distributed Ruby Send instance_eval/syscall Code Execution. Viitattu 1.4.2014. http://www.rapid7.com/db/modules/exploit/linux/misc/drb_remote_codeexec.

Rapid7 2014i. Samba "username map script" Command Execution. Viitattu 24.4.2014. https://www.rapid7.com/db/modules/exploit/multi/samba/usermap_script.

Rapid7 2014j. DistCC Daemon Command Execution. Viitattu 1.4.2014. https://www.rapid7.com/db/modules/exploit/unix/misc/distcc_exec.

Rapid7 2014k. Java RMI Server Insecure Default Configuration Java Code Execution. Viitattu 1.4.2014. http://www.rapid7.com/db/modules/exploit/multi/misc/java_rmi_server.

SANS 2006. What is Pen-Testing? Viitattu 18.3.2014. https://www.sans.org/reading-room/analysts-program/PenetrationTesting-June06.

Search Security 2007. Hijacking. Viitattu 13.2.2014. http://searchsecurity.techtarget.com/definition/hijacking.

Search Security 2014a. Buffer overflow. Viitattu 8.4.2014. http://searchsecurity.techtarget.com/definition/buffer-overflow.

Search Security 2014b. Distributed denial-of-service attack (DDoS). Viitattu 8.4.2014. http://searchsecurity.techtarget.com/definition/distributed-denial-of-service-attack.

Search Software Quality 2011. Pen test (penetration testing). Viitattu 13.2.2014. http://searchsoftwarequality.techtarget.com/definition/penetration-testing.

Secpoint 2014. Types of Hackers. Viitattu 7.2.2014 http://www.secpoint.com/types-of-hacker.html.

Sectools 2014. Nexpose. Viitattu 24.4.2014. http://sectools.org/tool/nexpose/.

Security Assessment 2014a. Internal Penetration Testing. Viitattu 13.2.2014. http://www.security-assessment.com/page/internal-penetration-testing.htm.

Security Assessment 2014b. External Penetration Testing. Viitattu 13.2.2014. http://www.security-assessment.com/page/external-penetration-testing.htm.

Social-Engineer 2014. Information Brokers. Viitattu 8.4.2014. http://www.social-engineer.org/framework/general-discussion/categories-social-engineers/information-brokers/.

Tenable 2014. Nessus Overview. Viitattu 31.3.2014. http://www.tenable.com/products/nessus.

Vacca J. 2013. Computer and Information Security Handbook. Steve Elliot.

Webopedia 2014a. Intrusion Detection System. Viitattu 8.4.2014. http://www.webopedia.com/TERM/I/intrusion_detection_system.html.

Webopedia 2014b. Social engineering. Viitattu 8.4.2014. http://www.webopedia.com/TERM/S/social_engineering.html.

Webopedia 2014c. Traceroute. Viitattu 3.4.2014. http://www.webopedia.com/TERM/T/traceroute.html.

Webroot 2014. What can computer hackers and predators do to me? Viitattu 20.2.2014. http://www.webroot.com/us/en/home/resources/articles/pc-security/computer-security-threats-hackers.

Western Carolina University 2014. Ten Tips For Protecting Your Computer From Hackers And Viruses. Viitattu 20.2.2014. http://www.wcu.edu/about-wcu/campus-services/university-police/campus-safety-crime-information/crime-prevention-information/ten-tips-for-protecting-your-computer-from-hackers-and-viruses.asp.

WineHQ 2014. Acunetix Web Vulnerability Scanner. Viitattu 31.3.2014. http://appdb.winehq.org/objectManager.php?sClass=application&iId=8379.

WISE 2014a. WISE Home. Viitattu 10.2.2014 http://wise.turkuamk.fi/?page=home.

WISE 2014b. WISE Overview. Viitattu 10.2.2014 http:// http://wise.turkuamk.fi/?page=overview.

WiseGeek 2014. In Computer Networking, What Is DMZ? Viitattu 8.4.2014. http://www.wisegeek.org/in-computer-networking-what-is-dmz.htm.

XSSF 2014. Overview. Viitattu 8.4.2014. https://code.google.com/p/XSSF/.